

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

**Progettazione e sviluppo di  
un'applicazione di realtà aumentata a  
supporto dell'insegnamento della  
modellazione 3D**

Applicazione ARhino



**Relatore:**

prof. Andrea Sanna

**Candidato:**

Gelsomino Monteverde

Dicembre 2018

Dedica...

# Ringraziamenti

Grazie

# Sommario

La didattica utilizza diverse strategie di insegnamento per formare gli individui in maniera da fargli acquisire le informazioni e le competenze specifiche di cui hanno bisogno. Questi metodi includono solitamente lezioni frontali in aula o libri di testo, questi però, possono essere integrati con metodi meno tradizionali che hanno un elevato potenziale, in quanto si basano su una maggiore interattività e dinamicità dell'apprendimento.

Uno strumento che può essere utilizzato è la realtà aumentata (AR). Questa è una tecnologia che permette di visualizzare, nell'ambiente, oggetti e informazioni virtuali in tempo reale e si basa sul principio dell'overlay. Per esempio, in un sistema mobile la fotocamera individua un marker, il sistema lo riconosce e renderizza un nuovo livello di comunicazione che si va a sovrapporre e a integrare alla realtà, "aumentando" la quantità di informazione. Questa tecnologia si è sviluppata notevolmente negli ultimi anni, grazie soprattutto alla forte evoluzione software e hardware, ed è oggi applicata in numerosi campi: in ambito militare per la simulazione di missioni, in ambito medico-riabilitativo per visualizzare direttamente sul corpo del paziente di modelli tridimensionali di organi e alterazioni patologiche (per esempio durante lo svolgimento di interventi chirurgici), in ambito industriale come sistema integrativo per la manutenzione e controllo dei processi di produzione ed infine come supporto all'insegnamento.

Questo lavoro di tesi si colloca in quest'ultimo ambito ed ha lo scopo di sfruttare l'AR come strumento di apprendimento. Difatti la realtà aumentata ha notevoli potenzialità nel campo dell'insegnamento in quanto permette di comprendere e modellare concetti astratti e complessi in maniera più semplice. Numerose teorie sull'apprendimento avvalorano questa tesi: su tutte il costruttivismo e la teoria dell'apprendimento situato.

Il primo è una teoria basata sull'idea che non è più l'insegnante al centro del processo di apprendimento, ma è il soggetto che apprende al centro di esso. Secondo questa teoria la conoscenza è intrinseca nel contesto dove il soggetto agisce ed è il prodotto di una costruzione attiva dello stesso, legata alla situazione concreta in cui avviene l'apprendimento.

L'idea dietro la teoria dell'apprendimento situato invece, è che l'acquisizione di informazioni avviene in un contesto specifico e che la qualità dell'apprendimento sia il risultato di interazioni tra le persone, i luoghi, i processi e la cultura all'interno e in relazione al contesto dato. L'uso dell'AR quindi, può produrre nuove prospettive di insegnamento in cui gli studenti prendono il controllo del proprio apprendimento e potrebbe fornire opportunità per stili di apprendimento e formazione più autentici. Come i risultati di diversi



studi hanno dimostrato, i sistemi AR possono fornire ambienti motivanti, divertenti e coinvolgenti che favoriscono l'apprendimento. Inoltre, le applicazioni AR in contesti educativi risultano attraenti e stimolanti per gli studenti. Un'altra teoria che avvalorava l'uso dell'AR è l'Edutainment, questa teoria si basa sul principio che l'educazione (educational) e il divertimento (entertainment), siano strettamente collegati e che un apprendimento efficace non prescinde da un atteggiamento positivo da parte dell'individuo.

Con queste premesse si è analizzato un problema rilevato nell'apprendimento di alcuni concetti di modellazione 3D, del corso di studi di "Design di scenario" della facoltà di Design. Per questo si è pensato di colmare questo deficit con lo sviluppo di un'applicazione mobile che utilizza la realtà aumentata. L'obiettivo di questa applicazione è quello di diventare uno strumento integrativo di insegnamento e supportare lo studente nell'apprendimento della materia. Prima di procedere allo sviluppo vero e proprio è stato analizzato il contesto di utilizzo dell'AR, ricercando progetti già realizzati nel campo specifico dell'apprendimento. L'obiettivo era quello di valutare, sulla base dei risultati ottenuti, se un progetto come quello pensato potesse avere un riscontro positivo e arrivare all'obiettivo prefissato. Dall'analisi degli articoli è stato rilevato che la maggior parte delle applicazioni di AR per l'educazione sono di tipo marker-based (che si basa sul tracciamento di un marker che contiene uno schema con colori molto contrastanti), questo perché attualmente il processo di tracking è molto più stabile rispetto a quello marker-less (che invece non ha bisogno di tracciare il marker per sovrapporre l'informazione). Anche la tecnologia location-based (che fa uso del GPS) è molto usata soprattutto rispetto a quella marker-less, questo grazie alla presenza di strumenti quali accelerometri, giroscopi, bussola e l'uso del GPS su dispositivi mobile. Inoltre si è riscontrato che l'efficacia della realtà aumentata emerge soprattutto in progetti che hanno come focus l'insegnamento di materie scientifiche quali la matematica, la geometria o la chimica. Infatti la possibilità di visualizzare oggetti (ad esempio i solidi o le molecole) permette di approcciarsi a problemi complessi in maniera più semplice. In particolare è stato notato che l'uso di questa tecnologia abbia notevole impatto sulle prestazioni di apprendimento ed induca lo studente ad avere una maggiore motivazione ad imparare e un "atteggiamento positivo" verso l'insegnamento. Gli unici svantaggi riscontrati nella ricerca sono relativi all'utilizzo di display e caschi troppo pesanti ed ingombranti da permettere un uso prolungato degli stessi e un iniziale disorientamento degli utenti dovuto all'uso di una nuova tecnologia per la prima volta.

Conclusa la fase di ricerca si è passati ad una fase esecutiva. In primo luogo sono stati definiti i requisiti base dell'applicazione, ovvero un'insieme di caratteristiche che si pensa possano permettere il raggiungimento degli obiettivi prefissati. L'applicazione è stata pensata come un gioco a livelli di difficoltà sempre maggiori, nei quali agli studenti viene richiesto di rispondere a delle domande sotto forma di quiz, che seguono gli argomenti delle slides del corso. Se l'utente dà una risposta corretta viene caricata la domanda successiva, mentre se la risposta è sbagliata viene permesso di riprovare ricaricando la stessa domanda, così da non scoraggiarlo ad andare avanti. Oltre alle domande l'applicazione permette di visualizzare le animazioni (in AR) relative ai passaggi spiegati nell'esercizio. Una volta terminata un'animazione viene richiesto di provare a replicare quanto visto, sul proprio computer con il software Rhinoceros. Infine è stato pensato un "sistema di punteggio" relativo ad ogni esercizio sotto forma di stelline, questo punteggio è calcolato in base

alle risposte esatte, quelle sbagliate e il tempo impiegato. Per ogni risposta esatta viene aggiunto un valore al punteggio finale, mentre ad ogni risposta sbagliata il punteggio viene decurtato di un valore (più piccolo). Infine in base al tempo realizzato il punteggio può subire un'ulteriore decurtazione. La logica generale dell'applicazione prevede un numero limitato di esercizi, che possono essere realizzati solo in maniera sequenziale, cioè per poter realizzare un determinato esercizio si deve aver completato quello precedente.

Una volta definita la struttura dell'applicazione, sono state selezionate le domande e le animazioni da realizzare, scegliendo i passaggi più significativi e difficili da comprendere relativi agli esercizi presenti nel materiale didattico già esistente. Le animazioni sono state realizzate utilizzando il software Blender, che ha permesso di creare sia gli oggetti 3D che le relative animazioni. In un secondo momento queste sono state trasferite in Unity 3D utilizzando il formato FBX. In seguito con il software Unity 3D è stata realizzata la logica generale dell'applicazione e sono state unite le animazioni precedentemente create. Infine il progetto è stato esportato come file APK per poter essere installato su dispositivi Android.

L'applicazione, denominata "ARhino", è stata implementata utilizzando il Model-View-Controller (MVC), il più usato nella programmazione object oriented e nello sviluppo di interfacce grafiche. Questo perchè rappresenta uno dei concetti fondamentali della programmazione ad oggetti e permette di strutturare l'applicazione in maniera molto efficiente; infatti, ogni classe viene progettata in modo da svolgere compiti specifici. Inoltre si è cercato di creare l'applicazione in maniera che sia scalabile, attraverso l'uso di files JSON (questi contengono i dati relativi alle domande e le animazioni) che permettono una modifica più veloce alla struttura in futuro.

L'applicazione verrà fornita agli studenti all'inizio del corso nel prossimo primo semestre come supporto all'insegnamento. La classe verrà divisa in due gruppi eterogenei (maschi e femmine): il gruppo sperimentale e il gruppo di controllo. Soltanto al primo verrà data la possibilità di utilizzare l'applicazione in aggiunta alle slides, mentre il secondo, ovvero gruppo di controllo seguirà il corso in maniera tradizionale. Alla fine del corso agli studenti del gruppo sperimentale verrà sottoposto un questionario per valutare le loro opinioni riguardo l'utilizzo di questa applicazione. Inoltre verrà fatta una comparazione tra i voti dei ragazzi presenti nei due gruppi, per analizzare se c'è stato un miglioramento nell'apprendimento e quindi nel rendimento in termini di votazione. Sulla base dei risultati del primo utilizzo si potrà poi pensare di accrescere le specifiche dell'applicazione. Ad esempio si potrà aumentare il numero di esercizi e creare un ambiente social per la condivisione delle informazioni e per incentivare la collaborazione tra gli studenti. Infine si potrebbe rendere i "premi" reali, come la possibilità di ottenere punti aggiuntivi all'esame finale, realizzando tutti gli esercizi con determinati punteggi.

# Indice

<b>Ringraziamenti</b>	II
<b>Sommario</b>	III
<b>1 Introduzione</b>	1
1.1 La Realtà Aumentata . . . . .	1
1.1.1 Cos'è . . . . .	1
1.1.2 Cenni Storici . . . . .	1
1.1.3 Principio di funzionamento . . . . .	4
1.2 Settori di utilizzo dell'AR . . . . .	5
1.2.1 Ambito militare . . . . .	5
1.2.2 Ambito medico-riabilitativo . . . . .	6
1.2.3 Manutenzione e controllo industriale . . . . .	6
1.2.4 Ambito training e learning . . . . .	6
1.3 Edutainment . . . . .	8
1.4 Realtà aumentata come strumento per l'insegnamento . . . . .	9
1.4.1 Situated Learning Theory . . . . .	10
1.4.2 Constructivism . . . . .	11
1.5 Obiettivi Tesi . . . . .	11
1.6 Struttura Tesi . . . . .	12
<b>2 Stato dell'arte</b>	13
2.1 Introduzione . . . . .	13
2.2 Studio università Girona . . . . .	15
2.2.1 Fase iniziale . . . . .	15
2.2.2 Risultati . . . . .	17
2.3 AR for Cultural Learning . . . . .	18
2.3.1 Risultati . . . . .	20
2.4 AR for Children Storytelling . . . . .	22
2.4.1 I due studi pilota . . . . .	23
2.5 AR for Mathematics and Geometry Education . . . . .	25
2.5.1 Risultati . . . . .	26
2.6 Sistema AR per Modellazione 3D . . . . .	26

2.6.1	Studio pilota . . . . .	27
2.6.2	Risultati . . . . .	28
2.7	AR for Situated Modeling with Tangible Primitive . . . . .	29
2.7.1	Studio . . . . .	30
2.7.2	Risultati . . . . .	30
2.8	AR Smartphone Application to Create 3D Models . . . . .	32
2.8.1	Studio . . . . .	33
2.8.2	Risultati . . . . .	34
2.9	Conclusioni . . . . .	34
<b>3</b>	<b>Tecnologie Utilizzate</b>	<b>35</b>
3.1	Blender . . . . .	35
3.1.1	Principali caratteristiche . . . . .	35
3.1.2	Animazione . . . . .	37
3.1.3	Shape Keys . . . . .	37
3.2	Unity3D . . . . .	38
3.2.1	Interfaccia e principali caratteristiche . . . . .	38
3.2.2	Animation Tool . . . . .	40
3.2.3	Scripting . . . . .	40
3.3	Vuforia . . . . .	41
3.3.1	Vuforia Scripting . . . . .	42
<b>4</b>	<b>Progettazione e sviluppo dell'applicazione ARhino</b>	<b>43</b>
4.1	Introduzione . . . . .	43
4.2	Gamification . . . . .	43
4.3	Struttura ARhino . . . . .	44
4.3.1	Flusso di Utilizzo . . . . .	44
4.3.2	Requisiti base . . . . .	48
4.3.3	Le Animazioni . . . . .	49
4.4	Architettura ARhino . . . . .	49
4.4.1	View ARhino . . . . .	51
4.4.2	Model ARhino . . . . .	51
4.4.3	Controller ARhino . . . . .	52
4.4.4	AR . . . . .	61
<b>5</b>	<b>Descrizione Esercizi</b>	<b>68</b>
5.1	Gli Esercizi . . . . .	68
5.1.1	Esercizio 1 . . . . .	68
5.1.2	Esercizio 2 . . . . .	74
5.1.3	Esercizio 3 . . . . .	75
5.1.4	Esercizio 4 . . . . .	75
5.1.5	Esercizio 5 . . . . .	75
5.1.6	Esercizio 6 . . . . .	76
5.1.7	Esercizio 7 . . . . .	76

5.1.8	Esercizio 8 . . . . .	76
5.1.9	Esercizio 9 . . . . .	77
5.1.10	Esercizio10 . . . . .	77
5.1.11	Esercizio 11 . . . . .	78
5.1.12	Esercizio 12 . . . . .	78
5.1.13	Esercizio 13 . . . . .	78
5.1.14	Esercizio 14 . . . . .	78
<b>6</b>	<b>Conclusioni e Sviluppi Futuri</b>	<b>80</b>
6.1	Conclusioni . . . . .	80
6.2	Sviluppi Futuri . . . . .	81
<b>A</b>	<b>Domande Esercizi</b>	<b>82</b>
<b>B</b>	<b>Questionario</b>	<b>95</b>
	<b>Riferimenti bibliografici</b>	<b>100</b>

# Elenco delle figure

1.1	Schema Mixed Reality. . . . .	2
1.2	Sensorama. . . . .	2
1.3	Il primo HMD progettato da Ivan Sutherland. . . . .	3
1.4	Un frame del gioco ARQuake. . . . .	4
1.5	Esercitazione militare con l'uso dell'AR. . . . .	5
1.6	Uso dell'AR durante un'operazione. . . . .	6
1.7	Applicazione in AR che assiste gli operai nel loro lavoro. . . . .	7
1.8	Elements4D: un' applicazione che permette di visualizzare i composti chimici. . . . .	7
2.1	Panoramica funzioni VECAR. . . . .	19
2.2	Esempi di VECAR nelle tre diverse attività. . . . .	20
2.3	Esempio di uso di due proiettori simultaneamente nella versione rivista di CoGAME. . . . .	22
2.4	Storytelling con l'uso di GENTORO. . . . .	23
2.5	Studente lavora con Construct3D, con un tutor e da solo. . . . .	25
2.6	Esempi di oggetti creati con 3DARModeler: (a) una piscina, (b) una chiesa, (c) una torre. . . . .	27
2.7	Esempio di modellazione situata con uso di primitive tangibili. . . . .	29
2.8	Esempio di traslazione e rotazione. . . . .	30
2.9	Modalità di creazione degli oggetti, partendo da semplici primitive per arrivare all'oggetto finale. . . . .	32
2.10	Immagine interfaccia utente dell'applicazione. . . . .	33
3.1	Interfaccia principale di Blender. . . . .	36
3.2	Interfaccia principale di Unity3D. . . . .	39
4.1	Schermata di introduzione dove vengono spiegate le istruzioni. . . . .	45
4.2	Schermata principale ARhino: il menu. . . . .	45
4.3	Schermata pausa. . . . .	47
4.4	Schermata fine esercizio. . . . .	48
4.5	Architettura Software ARhino. . . . .	49
4.6	Schema pattern MVC. . . . .	50
4.7	Convenzione nomi delle scene di tipo game. . . . .	55
5.1	Schermata domanda (in particolare la prima domande del primo esercizio). . . . .	69

5.2	Schermata risposta corretta [a], risposta sbagliata [b] . . . . .	69
5.3	Frame della prima animazione del primo esercizio (la prima metà della freccia è stata creata). . . . .	70
5.4	Schermata domanda successiva all’animazione: viene richiesto di mettere pausa. . . . .	71
5.5	Frame della seconda animazione del primo esercizio (si forma la freccia in 3D). . . . .	72
5.6	Schermata ultima domanda. . . . .	73
5.7	Diagramma a blocchi del primo esercizio (“ROUND 1”). . . . .	74

# Elenco dei codici

4.1	Classe Level.cs . . . . .	52
4.2	Metodo per il caricamento dati tra sessioni differenti. . . . .	53
4.3	Metodo per il salvataggio dati tra sessioni differenti. . . . .	54
4.4	Metodo per la visualizzazione delle domande e relative risposte. . . . .	56
4.5	Metodo per il calcolo finale del punteggio tenendo conto dei limiti di tempo. . . . .	57
4.6	Pezzo del metodo FillList() che permette di istanziare i bottoni relativi agli esercizi . . . . .	58
4.7	Pezzo del metodo FillList() che regola l'accensione delle stelline. . . . .	58
4.8	Metodo OnTrackingFound() richiamato ogni volta che la camera individua un marker. . . . .	61
4.9	Metodo OnTrackingLost() richiamato ogni volta che la camera perde un marker. . . . .	62
4.10	Metodo OnTrackableStateChanged() richiamato ogni volta che c'è un cambiamento nello stato dell'oggetto tracciato. . . . .	63
4.11	Classe NOARCamera realizzata per disabilitare la camera AR. . . . .	67



# Capitolo 1

## Introduzione

### 1.1 La Realtà Aumentata

« Augmented reality is a technology which: combines real and virtual, interactive in real time, registered in 3D. »<sup>1</sup>(Azuma 1997).

#### 1.1.1 Cos'è

Nella definizione di Azuma, uno dei maggiori ricercatori nel campo della Augmented Reality (AR), questa è descritta come: "...a technology that allows computer-generated virtual imagery information to be overlaid onto a live direct or indirect real-world environment in real time", <sup>2</sup> [1] A differenza della Virtual Reality (VR), nella quale gli utenti hanno un'esperienza completamente immersiva di un ambiente virtuale, nell'AR l'ambiente è reale, ma esteso da informazioni virtuali. L'AR funziona come una sorta di ponte tra il reale e il virtuale in modo da creare una continuità tra essi (Figura 5.7). In applicazione che utilizzano questa tecnologia il mondo reale prevale su quello virtuale.

#### 1.1.2 Cenni Storici

La storia dell'AR inizia intorno al 1960, il primo sistema funzionante usava un see-through display posizionato sulla testa dell'utente, il movimento del display poteva essere tracciato

---

<sup>1</sup> « La realtà aumentata è una tecnologia che: combina il reale e il virtuale, interattiva in tempo reale, registrata in 3D. »

<sup>2</sup> «...una tecnologia che permette di sovrapporre informazioni generate al computer nel mondo reale in tempo reale.»

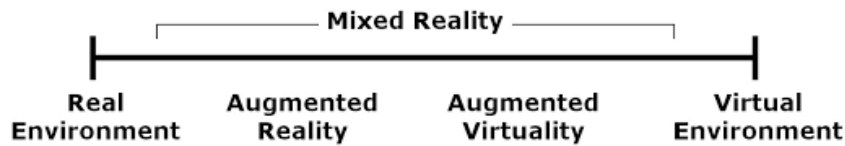


Figura 1.1: Schema Mixed Reality.

con due metodi diversi: attraverso un tracker di tipo meccanico o di tipo ultrasonico. Attraverso di esso si riuscivano a visualizzare soltanto dei semplici disegni in wireframe (un tipo di modello 3D nel quale sono visibili solo vertici e linee), a causa della limitata potenza dei computer in quegli anni.

Nel 1966 un fotografo, Morton Heilig brevettava un simulatore chiamato Sensorama. Questo era un dispositivo meccanico con il quale l'utente poteva vedere, ascoltare e perfino odorare cinque cortometraggi.

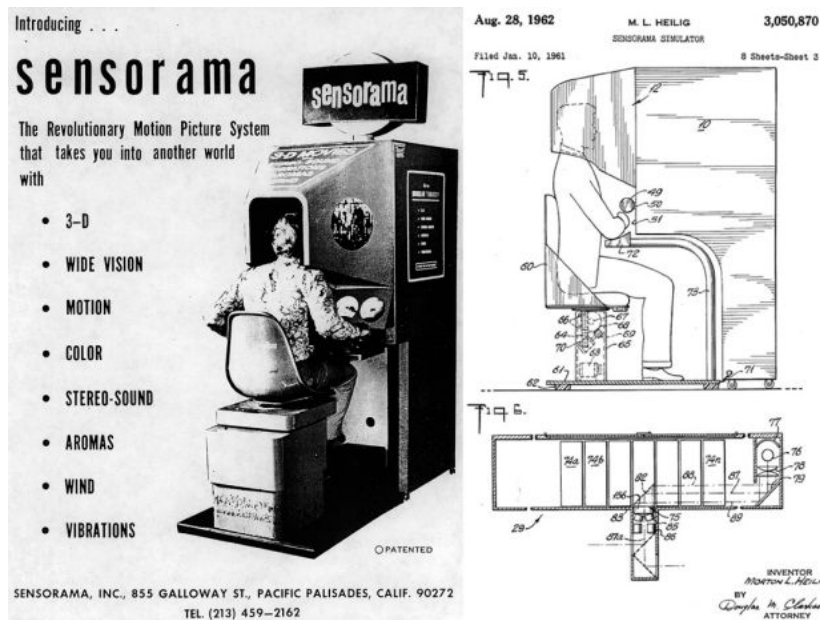


Figura 1.2: Sensorama.

Nel 1968 il professore e ricercatore ad Harvard, Ivan Sutherland, creava il primo Human mounted Display (HMD), denominandolo finestra sul mondo virtuale WoW (dall'inglese Window on the World). Questo apparecchio non era altro che un dispositivo da indossare in testa, spesso utilizzato come parte di un casco, che aveva un display ottico davanti ad uno o entrambi gli occhi.

Myron Kruger nel 1975 realizzava il primo dispositivo che permette di interagire con gli oggetti virtuali. Egli costruì un laboratorio di realtà virtuale nel quale più utenti potevano

interagire attraverso essa senza entrare in contatto diretto gli uni con gli altri e senza l'ausilio di occhiali o HMD.

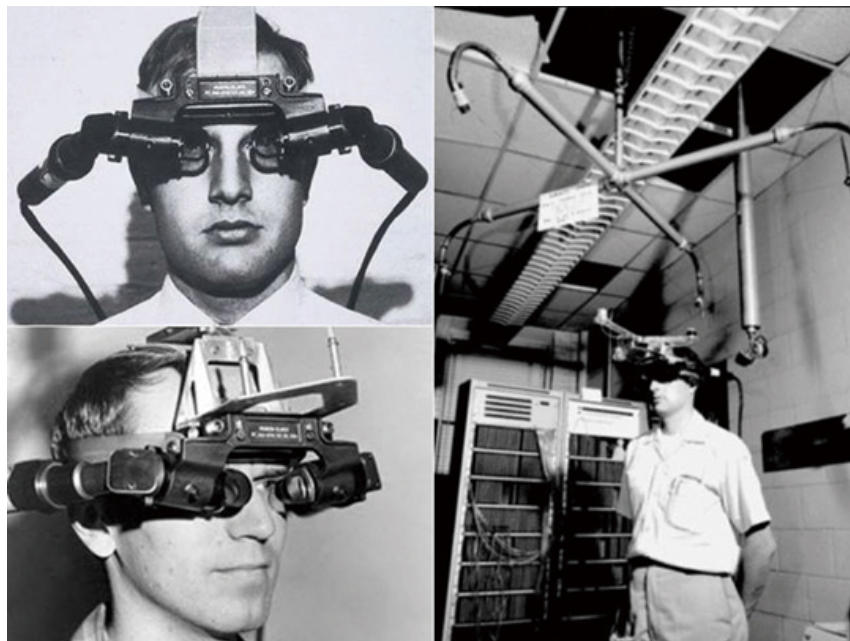


Figura 1.3: Il primo HMD progettato da Ivan Sutherland.

Il termine “realtà aumentata” viene attribuito all’ex ricercatore della Boeing Tom Caudell nel 1990. Nel progetto di Caudell veniva utilizzato un casco, sul quale erano montati due visori, questi permettevano di mostrare i dati virtuali relativi alle procedure di volo e atterraggio, senza però ostruire la visione del mondo reale. Ciò consentiva di “aumentare” (da qui il termine aumentata) il mondo reale con informazioni aggiuntive.

Nel 1994 Julie Martin fondò l’Augmented Reality Theater production, la prima compagnia teatrale nella quale i ballerini e acrobati si esibiscono in un mondo virtuale interagendo con esso. Nel 1997 Azuma introdusse l’attuale definizione d’AR e nello stesso anno venne creata la Touring Machine (sviluppata dalla Columbia University): questo dispositivo riusciva a mostrare all’utente informazioni addizionali, centrate sugli oggetti reali, riguardanti ciò che l’utente stesso stava guardando (ad esempio il nome degli edifici, delle strade, cenni storici legati ad essi).

A ridosso del 2000 è stata creata l’ARToolKit, una libreria che permette la creazione di applicazioni per l’AR attraverso il riconoscimento di immagini. Questa libreria è supportata da molte piattaforme sia mobile (Android) sia Web (Adobe Flash). [2] Sempre nel 2000 Bruce H. Thomas sviluppava ARQuake, versione in AR del famosissimo gioco Quake. Questo utilizzava un sistema GPS, un sensore di orientamento magnetico inerziale e ibrido, un controller che serviva da arma e un laptop inserito in uno zaino. Questo gioco non fu mai commercializzato e rimase solo un prototipo. Tuttavia questo portò un certo

interesse da parte delle aziende di games verso l'AR.[3]



Figura 1.4: Un frame del gioco ARQuake.

Nel 2003 Siemens SX1 rilasciava il primo gioco commerciale per dispositivi mobili in AR chiamato Mozzies. Il gioco consisteva nel colpire zanzare che venivano sovrapposte alla realtà attraverso la videocamera. Nel 2007 Reitmayr introdusse un sistema AR che permetteva di visualizzare informazioni legate ad edifici urbani ripresi attraverso un dispositivo mobile. [4]

### 1.1.3 Principio di funzionamento

Un sistema di realtà aumentata è costituito almeno da tre componenti principali che possono essere inseriti in tre sottoinsiemi:

- un generatore di scena;
- un dispositivo di visualizzazione delle immagini;
- un dispositivo di tracciamento.

Un quarto componente potrebbe essere un modello spaziale (ad esempio un database) per memorizzare le informazioni sul mondo reale e quello virtuale. Queste informazioni sono utili per determinare la posizione dell'utente nel mondo reale e per mantenere il contenuto di informazioni virtuale.

Un sistema d'AR può essere di due tipi: marker-based e mark-less based. Un sistema

marker-based, cioè basato sul riconoscimento di un marker bidimensionale, ha tre componenti principali: un “BOOKLET”(opuscolo) che fornisce le informazioni del marker; un “GRIPPER” che prende le informazioni e le converte in un altro tipo di dati e un “CUBE” che renderizza le immagini su uno schermo. Un sistema mark-less based, invece non necessita di marker, ma si basa su algoritmi di computer vision ed utilizza un sistema di tracking che coinvolge il GPS, una bussola e un dispositivo che permette il riconoscimento delle immagini.

## 1.2 Settori di utilizzo dell’AR

Negli anni, grazie alla forte evoluzione sia software che hardware, la potenza dei devices digitali è aumentata notevolmente. Questo ha permesso una evoluzione dei sistemi d’AR che oggi sono utilizzati in numerosi campi e forniscono molteplici soluzioni. Grazie alla sua versatilità l’AR può essere utilizzata sia in applicativi per tutti, utilizzati nella vita reale, sia in applicazioni di precisione pensate per settori specifici e professionali.

### 1.2.1 Ambito militare

Nel campo militare le applicazioni d’AR riguardano per lo più simulazioni di missioni, per esempio attraverso la proiezione di target virtuali nell’ambiente reale o attraverso la sovrimpressione in tempo reale sugli oggetti reali di informazioni relative all’ambiente e di possibili pericoli. In questo ambito l’uso d’AR permette di eliminare il pericolo di incidenti in situazioni reali e di abbassare i costi di esercitazione. I dispositivi più usati in questo campo sono i see-through HMD che non limitano molto il movimento.



Figura 1.5: Esercitazione militare con l’uso dell’AR.

### 1.2.2 Ambito medico-riabilitativo

Questo è uno dei campi in cui sono stati fatti i maggiori passi avanti poiché l'uso dell'AR permette delle pratiche innovative; per esempio è possibile visualizzare direttamente sul corpo del paziente modelli tridimensionali di organi e alterazioni patologiche utili soprattutto durante lo svolgimento di interventi chirurgici. Di recente sono stati realizzati studi sulla cura di fobie e riabilitazione, con l'uso d'AR immersiva, mediante la sovrapposizione nello spazio reale di punti di riferimento virtuali che guidano pazienti con mobilità limitata. In questi casi la caratteristica più importante di queste applicazioni è quella di definire uno spazio protetto per esplorare nuovi comportamenti e nuove simulazioni, superando paure e recuperando capacità perse.



Figura 1.6: Uso dell'AR durante un'operazione.

### 1.2.3 Manutenzione e controllo industriale

Nel campo aziendale ed industriale le applicazioni d'AR hanno permesso di limitare i costi di manutenzione e controllo delle grandi aziende. Infatti mentre prima questo tipo di lavoro veniva svolto da operai altamente specializzati, grazie all'AR anche gli operai non specializzati possono effettuare dei lavori di manutenzione su pezzi di grossi macchinari, guidati dalle informazioni che vengono fornite in modo virtuale. Anche in questo caso il device più usato è l'HMD per visualizzare informazioni che vengono sovrimpresse alle apparecchiature reali.

### 1.2.4 Ambito training e learning

L'ambito training è molto vasto, va dall'allenamento sportivo a quello industriale, aeronautico ecc... l'AR permette di ricreare dei contesti difficilmente realizzabili nella realtà e di guidare gli utenti in compiti specifici.



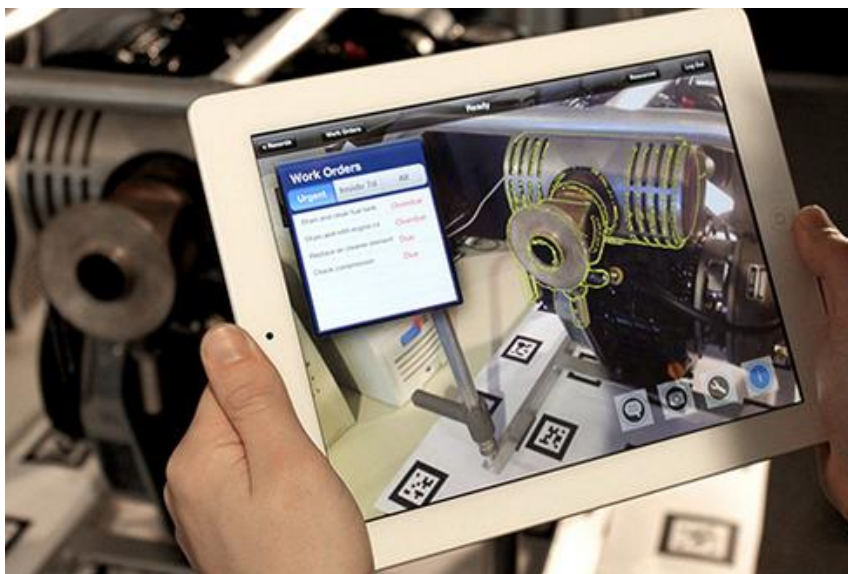


Figura 1.7: Applicazione in AR che assiste gli operai nel loro lavoro.

Nell'ambito learning si possono distinguere due ambiti in cui l'AR viene usata allo scopo di apprendere: nelle scuole come supporto all'insegnamento di diverse materie quali matematica, chimica, biologia, fisica; nelle aziende con lo scopo di far apprendere nuove capacità ai dirigenti e dipendenti. L'uso della realtà aumentata in questo ambito ha portato numerosi aspetti positivi riguardanti il grado di concentrazione degli studenti e la possibilità di usare canali di apprendimento più efficaci.

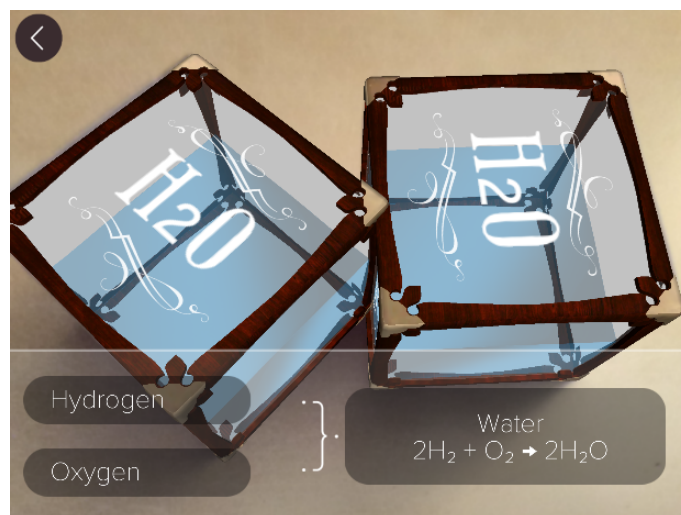


Figura 1.8: Elements4D: un' applicazione che permette di visualizzare i composti chimici.

## 1.3 Edutainment

Quando si parla di insegnamento, soprattutto attraverso l'uso di una nuova tecnologia come l'AR, non si può non introdurre un concetto fondamentale quale l'edutainment. Il termine edutainment deriva dall'unione di due termini anglosassoni educational (educativo) e entertainment (intrattenimento, gioco). M. McLuhan, grande sociologo dei media, spiega in una frase l'importanza di unire queste due pratiche: "Anyone who tries to make a distinction between education and entertainment doesn't know that the education must be amusing and the amusement must be educational."<sup>3</sup> [ M. McLuhan, (1964)]. Quindi partendo dall'idea che le persone imparano "giocando", numerosi studiosi di questo campo hanno sperimentato modalità di insegnamento e apprendimento inedite, create attraverso la sinergia fra il processo di formazione e il gioco, sfruttando i vantaggi legati all'interattività, alla multimedialità, alla connettività del mondo digitale.

La televisione ha da sempre avuto nel proprio palinsesto programmi con scopi educativi, prevalentemente rivolti a bambini, che permettessero non solo di intrattenere ma di educare alla manualità, all'uso di colori, alla conoscenza dell'alfabeto e dei numeri (esempi di programmi italiani: la Melevisione, l'Albero Azzuro e Art Attack).

Con l'avvento e l'evoluzione delle tecnologie digitali il discorso sull'edutainment si è spostato verso il computer e ancora più di recente verso lo schermo dei devices mobili. Nello specifico i videogames hanno portato nuove possibilità di apprendimento "divertente". Possiamo definire il computer e lo smartphone come media freddi nella definizione di M. McLuhan "...cool media imply a high degree of participation or completion by the public"<sup>4</sup>, [5] poiché internet e i videogames richiedono una partecipazione basata sull'interattività che prevede uno scambio comunicativo in cui ad uno stimolo deve corrispondere una risposta. Quindi come affermava lo studioso Gee "Players are pro-ducers, not just consumers; they are "writers" not just "readers". Even at the simplest level, players co-design games by the actions they take and the decisions they make"<sup>5</sup> [6], inoltre i videogames permettono la gratificazione ad obiettivo raggiunto che è fondamentale per mantenere attivo l'utente. Il gioco è uno strumento fortissimo che favorisce l'apprendimento grazie alla stimolazione delle emozioni che esercitano numerosi effetti sull'apparato cognitivo. Le emozioni infatti sono il motore dell'azione, una determinata emozione predispone il soggetto ad un'azione successiva che permette di far fronte ad un problema, a raggiungere un obiettivo. Naturalmente l'aumento di emozioni di tipo positivo favorisce l'autostima e la sicurezza e permette all'individuo di continuare nel percorso di apprendimento intrapreso, viceversa quelle negative possono bloccare immediatamente i progressi fatti.

---

<sup>3</sup> « Coloro che fanno distinzione tra intrattenimento ed educazione forse non sanno che l'educazione deve essere divertente e il divertimento deve essere educativo »

<sup>4</sup> « ...i media freddi implicano un alto grado di partecipazione o completamento da parte del pubblico »

<sup>5</sup> « I giocatori sono produttori non solo consumatori; sono "scrittori" non solo "lettori". Anche al livello più semplice, i giocatori co-progettano i giochi attraverso le azioni che compiono e le decisioni che prendono »



I videogiochi sono un medium dalle numerose sfaccettature e permettono nuove prospettive di insegnamento, non soltanto quelli pensati appositamente per l'apprendimento. Quello che è interessante è capire come l'uso a fini didattici possa portare allo sviluppo di nuove capacità intellettive e atteggiamenti indotti. Una delle caratteristiche che ha cambiato questo medium è la percezione della “morte”: con la logica del salvataggio del gioco e la possibilità di riprovare è possibile riprendere dal punto in cui si era arrivati senza dover ricominciare d'accapo. Per molti studiosi e psicologi questa opportunità è una caratteristica positiva che porta uno stimolo alla crescita, alla capacità di controllare e gestire autonomamente il sé, proprio in virtù della ripetizione continua dell'esperienza senza insoddisfazioni da parte del giocatore. Molto importate nei videogiochi è anche l'architettura stessa. Le diverse strutture hanno delle implicazioni sull'esperienza di gioco vera e propria. Esistono numerosi tipi di struttura: per esempio l'organizzazione a livelli, che divide l'esperienza in situazioni con difficoltà sempre maggiori è una di quelle più utilizzate. Un'altra caratteristica che influenza l'esperienza è l'agentività (agency in inglese), che Janet Murray definisce come: “The satisfying power to take meaningful action and see the results of our decisions and choices”<sup>6</sup>[7]. L'utente giocatore effettua delle scelte che si riflettono sul mondo circostante e lo modificano.

Oggigiorno le strutture dei videogiochi hanno un grado di agentività che varia notevolmente tra i vari giochi. Questo ha portato, e continua a farlo, ad un mutamento nel modo in cui i giocatori reagiscono, diventando più flessibili e sensibili ai segnali circostanti. Il giocatore non pensa al gioco come un addestramento, un processo di apprendimento e questo diventa un vantaggio. In questo campo si possono distinguere due tipi di apprendimento:

- un apprendimento esplicito, che consiste nel capire e far propri i codici e i contesti narrativi;
- un apprendimento collaterale, costruito sull'acquisizione di competenze e attitudini relative all'evoluzione del gioco.

L'apprendimento collaterale non ricade solo nel gioco, ma ha ripercussioni anche nella vita. John Dewey riassume così l'importanza dell'apprendimento collaterale: “Forse il maggiore degli errori pedagogici è il credere che un individuo impari soltanto quel dato particolare che studia in quel momento. L'apprendimento collaterale, la formazione di attitudini durature o di repulsioni, può essere e spesso è molto più importante. Codeste attitudini sono difatti quel che conta veramente nel futuro.” [8].

## 1.4 Realtà aumentata come strumento per l'insegnamento

Come già accennato, l'AR è stata studiata anche come mezzo per l'insegnamento. La realtà aumentata ha notevoli potenzialità come strumento di apprendimento “to enable

---

<sup>6</sup> « Il potere soddisfacente di fare un'azione significativa e vedere i risultati delle decisioni e delle scelte »

students to see the world around them in new ways and engage with realistic issues in a context with which the students are already connected”<sup>7</sup> [9]. Esistono due forme d’AR a disposizione degli insegnanti:

- location aware (basata sulla posizione dell’utente attraverso il tracciamento GPS su smartphone)
- vision based (basata sull’uso di una camera che viene puntata verso un oggetto/-marker).

Questi due tipi di AR sfruttando diverse caratteristiche degli smartphone (es. GPS, fotocamera, tracciamento immagini e oggetti) permettono delle esperienze di apprendimento “immersivo” nell’ambiente fisico. L’immersione è “...the subjective impression that one is participating in a comprehensive, realistic experience”<sup>8</sup>[10]. I dispositivi attuali permettono diversi gradi di immersione digitale. Quanto più un’esperienza virtuale immersiva combina fattori operativi, simbolici e sensoriali minore sarà la consapevolezza dell’utente di trovarsi in un ambiente “digitalmente aumentato”. Molti studi hanno dimostrato che l’immersione in un ambiente digitale può migliorare l’istruzione in almeno tre modi:

- consentendo molteplici prospettive,
- l’apprendimento situazionale
- trasferimento di capacità.

L’affermazione che l’AR riesca a fornire esperienze di apprendimento migliorate è supportata da due contesti teorici interdipendenti: la situated learning theory (teoria dell’apprendimento situato) e il costruttivismo (costruttivismo).

### 1.4.1 Situated Learning Theory

L’idea dietro la teoria dell’apprendimento situato è che esso avvenga in un contesto specifico e che la qualità dell’apprendimento sia il risultato di interazioni tra le persone, i luoghi, gli oggetti, i processi e la cultura all’interno e in relazione a quel contesto dato.[11]. Il processo dell’apprendimento è co-costruito e partecipativo, un processo in cui tutti gli individui vengono “...transformed through their actions and relations in the world”<sup>9</sup> [12]. Questa dottrina si fonda ed estende altre teorie sull’apprendimento come quella dell’apprendimento sociale e quella dello sviluppo sociale. Queste teorie enfatizzano l’importanza della qualità dell’interazione sociale nel contesto di apprendimento.

Si ha apprendimento quando si ha il trasferimento di conoscenze, acquisite in determinati contesti, in altre situazioni. Ci sono due strade per misurare il trasferimento di capacità:

---

<sup>7</sup> «Per permettere agli studenti di vedere il mondo intorno a loro in nuovi modi e attirarli con problemi realistici in un contesto con il quale gli studenti sono già connessi»

<sup>8</sup> «...l’impressione soggettiva che si partecipa ad un’esperienza completa e realistica»

<sup>9</sup> «...trasformati attraverso le loro azioni e relazione nel mondo.»

singole risoluzioni di problemi e preparazione per insegnamenti futuri[13].

Per esempio dare agli studenti delle istruzioni generali per risolvere un problema per poi far testare le loro abilità nel risolvere problemi simili, questo fa' sì che venga attivato il near-transfer: la capacità di applicare la conoscenza acquisita in una determinata situazione per un contesto simile con qualche caratteristica di superficie diversa. Quando invece si tratta di valutare il successo dell'apprendimento come preparazione per apprendimento futuro i ricercatori misurano il trasferimento su prestazioni estese in cui gli studenti "imparano ad imparare" in un ambiente ricco e risolvono problemi correlati in contesti reali. Con l'istruzione convenzionale e la soluzione di problemi semplici, la preparazione per l'apprendimento futuro richiede un trasferimento estremo: applicare le conoscenze apprese in una situazione in un contesto completamente diverso, la cui semantica sottostante è associata, ma distinta.[14]

### 1.4.2 Constructivism

Il costruttivismo è una teoria basata sull'idea che non è più l'insegnante al centro del processo di apprendimento, ma è il soggetto che apprende al centro di esso. Secondo questa teoria la conoscenza è intrinseca nel contesto dove il soggetto agisce ed è il prodotto di una costruzione attiva dello stesso, legata alla situazione concreta in cui avviene l'apprendimento e nasce dalla collaborazione sociale e dalla comunicazione interpersonale. Quindi la conoscenza del soggetto è basata su conoscenze pregresse e credenze che ogni individuo ha maturato nella sua vita, questa viene delineata dal livello di sviluppo, dalle proprie esperienze e il proprio background culturale.[15][16]

La realtà aumentata, come strumento cognitivo, si adatta bene con le teorie dell'apprendimento situato e il costruttivismo. Infatti lo studente è collocato in un contesto fisico e sociale reale, mentre viene guidato, agevola i processi di apprendimento partecipativi e metacognitivi come l'indagine e l'osservazione attiva, la collaborazione tra coetanei, l'insegnamento reciproco e la partecipazione legittima periferica con diversi modi di rappresentazione.[17]

## 1.5 Obiettivi Tesi

L'obiettivo principale di questa tesi è quello di realizzare un'applicazione in realtà aumentata, che supporti gli studenti, che seguono un primo corso di modellazione 3D, nell'apprendimento delle tecniche di modellazione mediante curve e superfici parametriche usate da software come Rhinoceros. Inoltre si cerca di comprendere se questa nuova tecnologia sia efficace in questo specifico esempio di learning e possa quindi diventare lo strumento principale nell'insegnamento del corso "Design di Scenario" nel Corso Di Laurea In Design E Comunicazione Visiva a Torino.

## 1.6 Struttura Tesi

Il presente lavoro di tesi è stato strutturato come segue: nel secondo capitolo c'è lo “stato dell'arte”, in cui vengono analizzati alcuni progetti inerenti all'AR e le applicazioni che ne fanno uso con scopo di istruire.

Nel terzo capitolo saranno presentate le tecnologie e gli strumenti utilizzati per la realizzazione del progetto. Ci sarà una panoramica dei software e di alcune loro funzionalità, usate nello svolgimento dell'applicazione.

Nel quarto verrà mostrato il progetto vero e proprio. In questa parte verranno descritte in modo accurato le metodologie con le quali il progetto è stato sviluppato, in che ordine e in che modo tutti i passaggi siano stati realizzati e mediante quali tecniche. Evidenziando eventuali difficoltà riscontrate durante il lavoro e come queste siano state superate.

Nel quinto capitolo verranno discussi i risultati del questionario sottoposto agli studenti. La parte finale di questo documento presenterà le conclusioni e verranno proposti degli sviluppi futuri per migliorare l'applicazione.

## Capitolo 2

# Stato dell'arte

### 2.1 Introduzione

«...the research in this field [AR] should continue and should be addressed to discover the affordances and characteristics of AR in education that differentiate this technology from others. Deepening this analysis will allow for discovering the unique value of the learning environments based on AR.»<sup>1</sup> [18][19]

Negli ultimi anni sono andate sempre ad aumentare le ricerche inerenti all'uso delle nuove tecnologie per migliorare l'insegnamento. Tra queste tecnologie una in particolare ha avuto una grande evoluzione: la realtà aumentata. Come già accennato, gli studi in questo campo sono stati favoriti dall'innovazione tecnologica, sia in campo hardware che software, dei dispositivi mobile e del loro crescente utilizzo giornaliero tra gli individui. Sono stati pubblicati numerosi articoli e libri sull'applicazione dell'AR nel campo educativo per diversi ambiti di apprendimento. La maggior parte della letteratura ha analizzato fattori quali: uso, scopi, vantaggi, limitazioni, efficacia e vantaggi dell'AR dal punto di vista dell'apprendimento. Una delle caratteristiche maggiormente evidenziata è l'immensa efficacia di questa tecnologia nel motivare lo studente nel processo di apprendimento. Tuttavia lo stato della ricerca attuale è ancora in fase iniziale rispetto al potenziale intrinseco dell'AR.

Analizzare la letteratura permette di comprendere appieno le potenzialità e di proporre nuove tendenze, sfide, opportunità per nuove ricerche, per esempio la possibilità di adattamento e personalizzazione nelle applicazioni educative AR o ancora lo sfruttamento di questa tecnologia per necessità particolari degli studenti in contesti diversi.

---

<sup>1</sup> «la ricerca in questo campo dovrebbe continuare e dovrebbe essere indirizzata a scoprire le affordances e le caratteristiche dell'AR nell'istruzione che differenziano questa tecnologia da altre. Approfondire questa analisi permetterà di scoprire il valore unico degli ambienti basati sull'AR»

Negli ultimi decenni alcuni studiosi ed insegnanti hanno sviluppato delle applicazioni in AR come strumenti da affiancare alle tecniche classiche.

Nell'ambito della scuola primaria e secondaria nel 2008 Freitas & Campos svilupparono SMART (system of augmented reality for teaching), questo sistema usava l'AR per insegnare concetti molto semplici come i mezzi di trasporto e gli animali. Esso prevedeva la sovrimpressione di modelli tridimensionali e prototipi come un aereo, una macchina, un camion su video in tempo reale mostrato a tutta la classe. Freitas & Campos eseguirono numerosi esperimenti su 54 studenti di tre scuole diverse in Portogallo. I risultati indicavano come SMART aiutava ad aumentare la motivazione tra gli studenti e aveva un impatto positivo sull'esperienza di apprendimento soprattutto tra gli studenti meno bravi. Per quanto riguarda l'istruzione secondaria e l'università è stato studiato come l'AR aiuti a migliorare la conoscenza e le capacità soprattutto per quanto riguarda teorie complesse o meccanismi di sistemi o macchinari. Per esempio Liarokapis (2004) ha verificato come usando l'AR gli studenti riuscissero ad assimilare concetti e meccanismi complicati facendo dei test su un modello 3D di albero a canne associato ad un insieme di componenti reali del motore.

Di seguito verranno descritti alcuni esempi di applicazione dell'AR in differenti materie. Nelle classi di astronomia gli insegnanti usano modelli 3D dei pianeti e delle stelle per rendere trasparente la relazione tra essi. Un esempio arriva dallo studio di Shelton(2002) egli aveva riscontrato come la possibilità di vedere e manipolare i pianeti e il sole virtuali attraverso una piattaforma tenuta in mano che permetteva di cambiare le coordinate e l'orientamento, consentiva agli studenti di capire in maniera profonda come i vari elementi si muovevano relativamente agli altri. Un altro esempio è lo SkyMap di Google in cui numerose informazioni sono sovrimpresse con la tecnologia dell'AR sulle costellazioni che gli utenti inquadrano con la fotocamera del cellulare.

Nell'insegnamento della chimica un esempio è "Augmented Chemistry" che consiste in un banco di lavoro che permette di visualizzare gli atomi e le molecole e studiare la loro composizione. Esso è composto da un "booklet" che contiene i marker per ogni elemento, un "gripper" con un pulsante che permette di collegare l'atomo al modello molecolare e un "cube" che viene ruotato dagli utenti per determinare come e dove l'elemento si collega alla molecola.

In biologia l'AR può essere usata per visualizzare la struttura e l'anatomia del corpo. The specialist Schools and Academies Trust (SSAT) ha dimostrato che l'AR può aiutare gli studenti a capire la struttura dei singoli organi umani attraverso la visione direttamente in classe di modelli 3D generati al computer.

Nel campo della matematica e geometria l'AR favorisce e facilita l'apprendimento di argomenti a volte molto complessi. Grazie alla realtà aumentata gli studenti e gli insegnanti possono collaborare interagendo su problemi riguardanti figure e la combinazione tra più di esse. Un esempio di applicazione è Construct3D, realizzata specificatamente per l'insegnamento geometrico con l'uso di modelli di figure 3D, la quale permetteva di condividere uno spazio virtuale in maniera collaborativa per costruire forme geometriche con l'uso di un HMD.

Nell'insegnamento della fisica l'AR può essere usata per esempio per dimostrare le proprietà cinematiche. Duarte, Cardoso & Lamounier Jr. (2005) valutarono l'AR per rappresentare dinamicamente un oggetto che varia nel tempo come la velocità e l'accelerazione. L'uso della realtà aumentata permetteva di visualizzare i risultati reali e quelli stimati e di migliorare l'apprendimento.

Interessante è anche l'utilizzo dell'AR su uno strumento più tradizionale di apprendimento: il libro. Billinghamurst (2001) ha studiato come le persone soprattutto i più piccoli possano leggere un libro in una maniera più interattiva e realistica sovrapponendo modelli renderizzati sui libri. Un esempio di applicazione di questo genere è il cosiddetto "Magic Book" (una versione aumentata dei libri 3D "pop-up") che fa uso di un visore tenuto in mano che permette di entrare nella storia raccontata dal libro e sentirsi parte di essa.

## 2.2 Studio università Girona

Uno studio del 2014 dell'università di Girona ha raccolto numerose riviste e articoli per indagare i trends dell'AR nel campo dell'insegnamento.

### 2.2.1 Fase iniziale

Per iniziare è stato utilizzato l'indice di Google Scholar usando come categoria "tecnologia educativa", in seguito sono state indicizzate le riviste relative alla tecnologia dell'istruzione insieme alle riviste riguardanti la ricerca educativa in generale. Sono state scelte le prime cinque riviste dalla categoria "tecnologia educativa" che sono andate a formare una lista ("lista GS"). Per convalidare questa lista iniziale, è stato eseguito un processo di duplice verifica iterativo utilizzando lo strumento JCR SSCI (Journal Citation Report Social Science Citation Index), che permette di analizzare il fattore di impatto di ogni giornale e la sua "correlazione" con gli altri. La "correlazione" è definita nel JCR e si basa sul numero di citazioni da una rivista all'altra e il numero totale di articoli. Sono stati individuati criteri di inclusione/esclusione e categorie di ricerca ben definite. Criteri di inclusione:

- studi pubblicati dal 2003 al 2013 (inclusi gli articoli scritti fino a Febbraio 2014);
- studi che hanno come argomento applicazioni di realtà aumentata nell'insegnamento;
- studi che approfondiscono vantaggi, svantaggi, limitazioni, caratteristiche, usi, sfide l'efficacia della realtà aumentata nei contesti educativi;
- studi che descrivono le applicazioni di realtà aumentata nell'istruzione per le persone in contesti diversi;
- studi che descrivono i metodi di valutazione delle applicazioni di realtà aumentata negli scenari educativi;

Criteri di esclusione:

- studi non identificati come "articoli" (ad esempio recensioni di libri, libri ecc.);

- studi che menzionavano l'AR ma non avevano come argomento principale la realtà virtuale o altri argomenti.

Categorie di ricerca:

- campo di istruzione: sulla base della classificazione internazionale standard dell'educazione (UNESCO, 2012);
- gruppo di destinazione: sulla base della Classificazione Standard Internazionale dell'Istruzione (UNESCO, 2012);
- scopi riportati sull'utilizzo di AR;
- vantaggi dell'AR riscontrati;
- limiti dell'AR;
- efficacia dell'AR;
- tipo di AR utilizzata;
- tipo di processo di adattamento;
- bisogni speciali riscontrati;
- metodo di intervento;
- campione di ricerca;
- metodo di ricerca;
- dimensione temporale;
- metodi di raccolta dati.

Utilizzando questi criteri sono stati selezionati 32 studi dalle riviste, su questi sono stati raccolti i dati e analizzati allo scopo di comprendere i passi in avanti fatti nel campo dell'AR applicata all'insegnamento e definire nuovi metodi per il futuro. Dall'analisi è emerso che negli ultimi anni il numero degli studi su questo argomento sono aumentati anno dopo anno. Da questo si può dedurre che molti ricercatori sono interessati ad esplorare le caratteristiche, i vantaggi e i limiti dell'AR in campo educativo. Come Bujak dice: "La realtà aumentata ha appena iniziato a scalfire la superficie nell'applicazione in campo educativo".



### 2.2.2 Risultati

I risultati raccolti hanno evidenziato come il 40.6% degli articoli riguarda l'applicazione dell'AR all'insegnamento delle scienze. Una possibile spiegazione è che l'AR è molto utile per spiegare concetti che non sono visibili nel mondo reale o con dispositivi specializzati. Quindi la possibilità di assimilare concetti astratti e complessi che nella realtà sarebbero molto difficili da comprendere.

Al secondo posto troviamo il campo umanistico e delle arti con il 21.9%. Gli studi in questo ambito si focalizzano sull'insegnamento delle lingue, arti visive e culturalismo. Per l'insegnamento delle lingue e le arti visive l'AR è stata largamente usata poiché permette da una parte di offrire informazioni “aumentate” combinate con contesto reale e dall'altra la possibilità di una esperienza visiva migliorata. Con il 15.6% il campo dell'ingegneria per manufatti e costruzioni. Il campo meno analizzato è quello della salute e benessere con il 3.1%.

Per quanto riguarda il gruppo di riferimento, che si riferisce al livello di istruzione dei partecipanti agli esperimenti, lo studio ha evidenziato come l'AR è stata applicata soprattutto in ambienti di istruzione superiore (laurea ed educazione obbligatoria). I risultati più significativi di questi studi enfatizzavano come gli individui che partecipavano agli esperimenti erano più motivati ad apprendere e, come l'uso dell'AR, facilitava la spiegazione e quindi l'apprendimento di alcuni argomenti. Un altro dato importante emerso è il fatto che nessuno degli studi avesse come target i bambini, questo può essere spiegato dalla difficoltà che una tecnologia come la realtà aumentata può aggiungere al momento dell'apprendimento (per esempio la necessità di una forte interazione con il device e l'uso di marker per il tracking).

Un'altra analisi molto significativa è quella riguardante lo scopo dell'uso dell'AR nel campo dell'educazione. I dati evidenziano come la maggior parte degli studi usano questa tecnologia per spiegare un argomento (43.7%) e per accrescere delle informazioni (40.6%). Inoltre viene rilevato come una buona percentuale di studi riguardi il suo utilizzo in giochi educativi (18.7%) o per esperimenti di laboratorio (12.5%); questi due campi hanno numerose possibilità di sviluppo, basti pensare alla riduzione dei costi per quanto riguarda gli esperimenti in laboratorio e la possibilità di offrire un'esperienza più inclusiva verso le persone con disabilità.

Ulteriori elementi analizzati dallo studio di Jorge Bacca sono i “vantaggi riscontrati” negli studi approfonditi. Anche in questo caso ogni studio può avere più vantaggi. I risultati indicano come siano due i vantaggi maggiormente riscontrati: aumento dell'apprendimento (43.7%) e la motivazione (31.2%). Questi dati ancora una volta confermano i vantaggi dell'uso dell'AR per migliorare l'apprendimento e motivare gli studenti durante il processo. Un altro elemento emerso come vantaggio è la possibilità della collaborazione, che è una caratteristica intrinseca della tecnologia AR.

Inoltre sono state studiate anche le limitazioni di questa tecnologia. La limitazione maggiormente riscontrata riguarda la difficoltà di mantenere le informazioni sovrainpresse digitalmente (9.3%), gli studenti si possono scoraggiare se l'applicazione non funziona in

modo adeguato. Un'altra limitazione emersa è la necessità di dover prestare troppa attenzione alle informazioni virtuali (6.2%). Questo è dovuto soprattutto ad una fase iniziale in cui lo studente si trova ad avere a che fare con una nuova tecnologia, quindi le informazioni virtuali possono essere motivo di distrazione. Un'altra limitazione riscontrata è il fatto che in alcune occasioni l'AR è una tecnologia invasiva (6.2%) per esempio quando per fruire dell'applicazione c'è bisogno di devices come l'HMD.

Per quanto riguarda i dati sull'efficacia dell'AR come strumento per l'apprendimento molti studi hanno riportato come questa tecnologia porti a migliori prestazioni di apprendimento (53.3%), una maggiore motivazione ad imparare (28.1%) e un maggior impegno da parte degli studenti (15.6%). Meno riscontrati ma lo stesso molto importanti nel campo dell'insegnamento sono: "miglioramento del piacere percepito" (12.5%) e "atteggiamenti positivi" (12.5%). Infine sono stati analizzati i dati relativi al tipo d'AR utilizzata. Lo studio dell'università di Girona ha preso in considerazione tre tipi d'AR secondo la classificazione di Wojciechowski & Cellary:

- marker-based AR;
- marker-less AR;
- location based AR.

Come già accennato nel capitolo precedenti l'AR marked-based si basa sul tracciamento di un marker che contiene uno schema con colori molto contrastanti (tipo bianco e nero), la tecnologia mark-less based si basa sul riconoscimento della forma di un oggetto, mentre l'AR location based si fonda sulla geolocalizzazione dell'utente. I risultati degli studi hanno rilevato che la maggior parte delle applicazioni di AR per l'educazione sono di tipo marker-based (59.3%), questo perchè attualmente il processo di tracking è molto più stabile rispetto a quello marker-less (12.5%), che viene comunque utilizzato con l'ausilio di strumenti quali il Kinect della Windows che ha un buon tracciamento degli oggetti. Infine è interessante notare come la tecnologia location-based sia molto usata (21.8%) soprattutto rispetto a quella marker-less, questo grazie alla presenza di strumenti quali accelerometri, giroscopi, bussola e l'uso del GPS su dispositivi mobile. Nei prossimi paragrafi verranno analizzati più in dettaglio alcuni casi studio approfonditi.

## 2.3 AR for Cultural Learning

"Computer-assisted culture learning in an online Augmented reality environment based on free-hand gesture interaction. 2014 Mau-Tsuen Yang and Wan-Che Liao."

Esistono numerosi ambienti di apprendimento virtuale costruiti per migliorare la comprensione culturale, le abilità di comunicazioni tra diverse culture e l'interazione tra persone di paesi diversi. Alcuni esempi sono: Croquelandia applicazione sviluppata all'università del Minnesota creata specificamente per imparare a fare delle richieste in spagnolo; Second China sviluppata nel 2008 all'università della florida che consisteva in un'isola virtuale in

Second Life creata per imitare gli aspetti culturali e visivi della Cina; Bi-Lateral un gioco di simulazione per la pratica di negoziazione in un contesto di cultura araba ecc. . .

In questo caso in particolare si parla di un sistema denominato VECAR che consiste nell' "aumentare" un'aula con algoritmi di AR e computer vision (CV), per insegnare l'inglese a studenti stranieri. Il VECAR (Figura 2.1) prevede la registrazione 3D della mano e dei suoi movimenti, del banco di lavoro e un'interazione intuitiva con essa. Ogni studente indossa un paio di occhiali, un display e una fotocamera montati in testa. Le immagini catturate dal vivo vengono elaborate da VECAR. Inoltre lo studente ha la possibilità di manovrare gli oggetti virtuali usando gesti della mano che permettono di combinare oggetti virtuali con un banco fisico. Questo sistema è stato testato in tre diverse attività (Figura 2.2) di apprendimento: un viaggio virtuale di Londra (LA1), un'esplorazione dell'America dell'Ovest basato sul libro "Seaman: The Dog Who Explored the West with Lewis and Clark" di G.Karwoski (LA2) , la terza attività introduceva lo sport più seguito degli USA: il football americano.

Lo scopo di questa ricerca era triplice. Il primo era di valutare l'efficacia dell'apprendi-

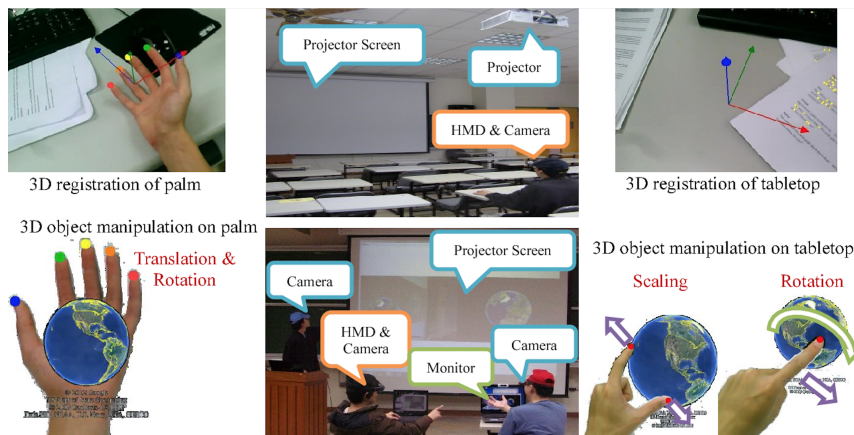


Figura 2.1: Panoramica funzioni VECAR.

mento degli studenti di contenuti culturali attraverso l'uso del VECAR; il secondo era di determinare i comportamenti di comunicazione tra insegnanti e studenti; infine il terzo obiettivo era di verificare l'utilizzabilità e la facilità d'uso di questo sistema.

L'esperimento è stato effettuato su 44 studenti (tra i 22 e i 26 anni) laureati in ingegneria, divisi in due gruppi: il gruppo sperimentale e il gruppo di controllo (divisi ugualmente). La valutazione è stata divisa in fasi diverse: nella prima i partecipanti di entrambi i gruppi hanno fatto un pretest (fase di spiegazione delle funzionalità base del VECAR); nella seconda fase è stata condotta la vera e propria attività di apprendimento, nella quale tutti gli studenti erano stati dotati di un pc con accesso a internet, ma solo il gruppo sperimentale poteva accedere alle informazioni attraverso il sistema virtuale, mentre l'altro gruppo usava mouse, tastiera, Google Earth e altri materiali web; nell'ultima fase tutti e due i gruppi si sottoponevano a due questionari.

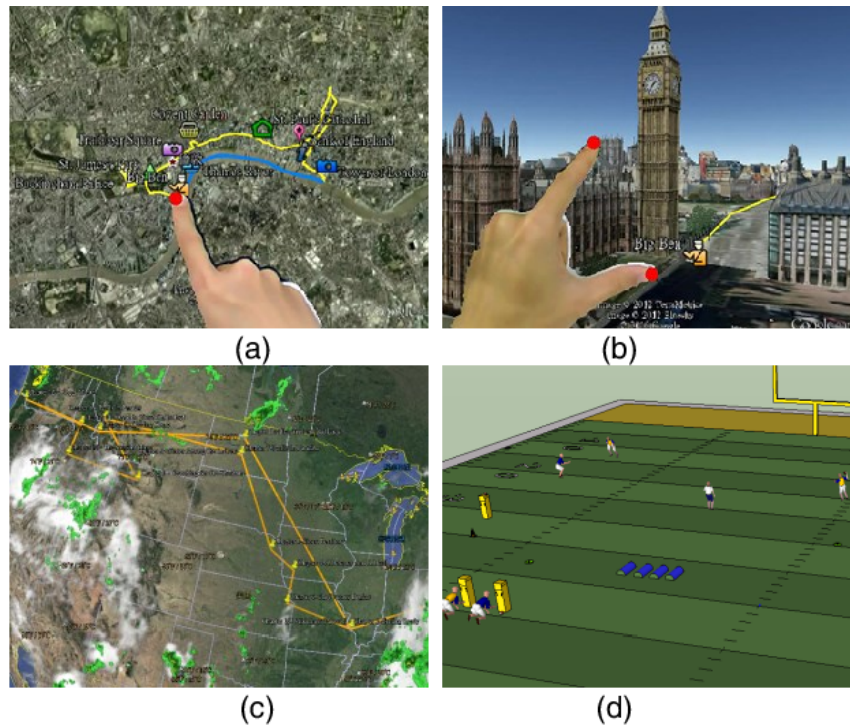


Figura 2.2: Esempi di VECAR nelle tre diverse attività.

### 2.3.1 Risultati

I punteggi rilevati dai test effettuati prima e dopo l'attività di apprendimento hanno evidenziato, in entrambi i gruppi, un miglioramento della conoscenza culturale, infatti si nota una notevole differenza di punteggio medio tra i test iniziali e quelli conclusivi (punteggio medio pre-test 38.2; punteggio medio post-test 78.6).

Inoltre sono stati analizzati i dati per capire la differenza di apprendimento nei due gruppi, questa analisi ha evidenziato una differenza non rilevante per quanto riguarda i pre-test con un valore di  $p$  maggiore di 0.05 (da questo si può dedurre che i due gruppi avessero le stesse conoscenze culturali). Mentre per quanto riguarda i dati dei post-test si evidenzia una maggiore differenza statistica con un valore di  $p$  minore di 0.05. Questi risultati indicano che l'efficacia dell'apprendimento culturale attraverso VECAR era migliore rispetto al gruppo di controllo che non aveva utilizzato il sistema virtuale. Questo soprattutto grazie alla possibilità data da VECAR di poter traslare, ruotare, ridimensionare e visualizzare oggetti 3D utilizzando dei gesti intuitivi della mano. Gli studenti erano in grado di visualizzare e giocare con nozioni culturali come ad esempio le regole del football americano che sono difficili da comprendere attraverso la sola spiegazione verbale.

Per quanto riguarda il secondo obiettivo i dati evidenziano come i punteggi medi del gruppo sperimentale fossero superiori a i dati corrispondenti al gruppo di controllo. In

particolare i risultati relativi alle categorie “non-verbal support” ( $M_{a-b} = -0.44$ ) e “challenging” ( $M_{a-b} = -0.44$ ). Nel complesso si osserva una differenza significativa nei punteggi tra il gruppo di controllo e il gruppo sperimentale. Questa considerazione implica che il VECAR aiuti molto la comunicazione interpersonale tra insegnante e studente. Il VECAR infatti permette agli studenti di muovere gli oggetti virtuali avanti e indietro tra il banco e lo schermo di proiezione, questo incoraggia la comunicazione tra insegnante e studente. Nell’analisi dell’apprendibilità ( $M = 73.30$ ) è stato riscontrato come questo aspetto abbia ottenuto un punteggio inferiore rispetto all’usabilità ( $M = 80.68$ ) di VECAR, questo fa pensare che sia necessario più tempo per la formazione e per prendere familiarità con l’interfaccia virtuale e gestuale. In generale comunque l’esperienza soggettiva è buona per chi ha usato VECAR (valore “overall” = 79.20). Dalle interviste realizzate alla fine dell’esperimento, tuttavia, sono emerse delle problematiche legate all’uso di VECAR: necessità di padroneggiare in poco tempo una nuova tecnologia, la sensibilità da parte del sistema alle occlusioni visuali a rumori di fondo e cambi di illuminazione che hanno come conseguenze perdita di concentrazione da parte degli studenti e rilevazioni errate. [20]

## 2.4 AR for Children Storytelling

“A Mobile Mixed-Reality Environment for Children’s Storytelling Using a Handheld Projector and a Robot. 2011 Masanori Sugimoto.”

Sono stati realizzati molti sistemi per supportare le attività di gioco e narrazione dei bambini. Per esempio Story Room che permette di creare ambienti interattivi in modo intuitivo in cui il bambino è autore e protagonista delle storie; Kipad che consente ai bambini di disegnare diverse scene di storie in maniera simultanea e in maniera molto semplice; Magic Carpet che “aumenta” il sistema Kipad con interfacce tangibili che fungono da supporto nel raccontare storie in un contesto scolastico.

In questo caso studio viene presentato un sistema chiamato GENTORO che a differenza degli altri sistemi prima elencati permette ai bambini di gestire la narrazione con l’utilizzo di un robot e dei proiettori portatili come se si stesse realizzando un film con effetti speciali creati in tempo reale. Queste caratteristiche fanno sì che ci sia una maggiore collaborazione tra i bambini e fa sviluppare in essi il coordinamento. Il sistema GENTORO deriva da un altro sistema precedente: CoGAME (Figura 2.3). Questo era un sistema per un’applicazio-

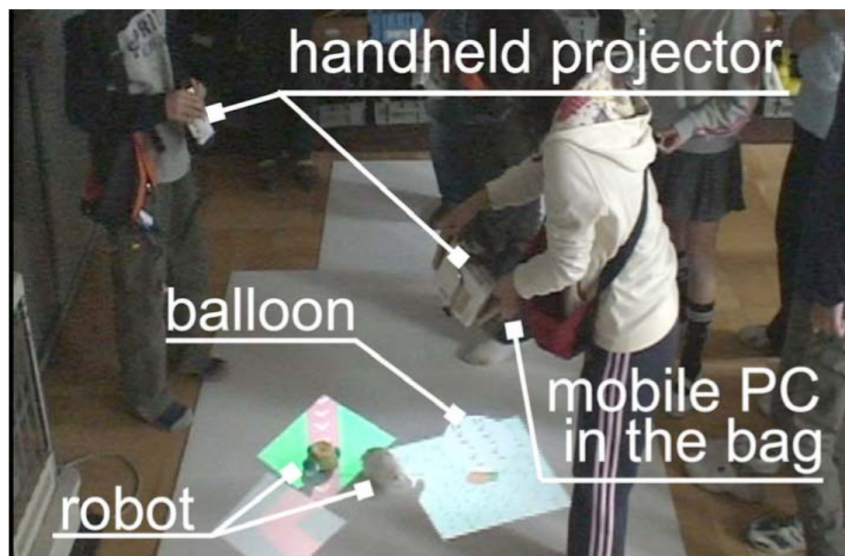


Figura 2.3: Esempio di uso di due proiettori simultaneamente nella versione rivista di CoGAME.

ne di intrattenimento che permetteva a più giocatori in modo cooperativo di manipolare un robot fisico senza l’ausilio di sensori e guidarlo a destinazione utilizzando un proiettore portatile. Furono identificate numerose limitazioni alla narrazione per bambini: si poteva controllare solo un robot, questo limitava la possibilità di fantasia e invettiva dei bambini, il dispositivo di controllo (un proiettore palmare) non era adatto a dei bambini e il sistema

CoGAME mancava di funzioni per processi di disegno e rendering della storia. Per questo furono effettuati due studi pilota che avevano lo scopo di evidenziare le caratteristiche positive e negative del sistema CoGAME così da poter definire i requisiti di progettazione più vantaggiosi per GENTORO (Figura 2.4).

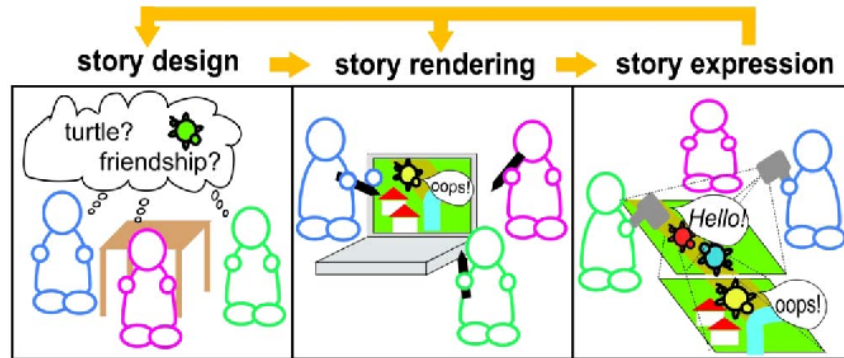


Figura 2.4: Storytelling con l'uso di GENTORO.

### 2.4.1 I due studi pilota

Il primo studio fu effettuato in una scuola del Giappone su 24 studenti divisi in 8 gruppi. Ogni gruppo doveva controllare due robot (una tartaruga e un coniglio) con il proiettore. Lo scopo in particolare era capire i problemi e le difficoltà quando i bambini usavano il proiettore e le eventuali soluzioni e i pro e i contro del controllo multi robot.

I risultati più evidenti emersi da questionari e analisi di video erano che la maggior parte degli studenti (92%) riusciva facilmente a capire come controllare e manipolare il robot; il 75% aveva un forte interesse nel far interagire i due robot. D'altra parte il sistema a volte falliva il riconoscimento tramite la fotocamera e non sempre permetteva il trasferimento del controllo da un proiettore all'altro; inoltre c'erano dei problemi di visualizzazione dovuti alla breve distanza tra proiettore e schermo/piano che rendeva le immagini troppo piccole.

Nel secondo studio pilota furono inseriti dei moduli software che permettevano i processi di rendering della storia. Questi includevano un modulo di disegno scena, un modulo per l'impostazione del percorso dei robot e uno di simulazione della storia. Questo studio venne realizzato in un solo giorno e furono inseriti 7 bambini che avevano partecipato anche al primo. I risultati hanno evidenziato come il modulo di disegno scena non era efficace a causa della bassa risoluzione del tablet usato, mentre il modulo di impostazione del percorso dei robot era molto semplice e utile da usare. Il modulo di simulazione di scena era efficace perchè permetteva ai bambini di specificare una durata per ogni scena, inoltre i bambini usavano oggetti reali come punti di riferimento per lo svolgimento della storia.



Da questi due studi furono specificati i requisiti software e hardware per il sistema GENTORO e fu fatto un esperimento per analizzare l'effettivo vantaggio dell'uso di questo sistema. Questo avvenne nel 2008 con 25 studenti con età tra gli 11 e 12 anni, divisi in 5 gruppi. Per prima cosa venne chiesto di creare una storia di una "tartaruga itinerante" e ad ogni gruppo fu assegnato un tema diverso. In una seconda fase venne richiesto di rivedere quella storia in base ai risultati della prima. I risultati emersi dalle indagini sui bambini hanno messo in risalto i pro e i contro di GENTORO: risposte positive sono state date sull'usabilità e anche sull'utilizzo del proiettore; la caratteristica mobile di questo sistema ha permesso di coinvolgere fortemente i bambini e di farli agire in modo coordinato tra di loro, inoltre è stato riscontrato come le storie che erano emerse dalla seconda fase erano tutte più logiche rispetto alla prima.

Mentre i contro erano tutti di tipo software, infatti i bambini hanno riscontrato problemi con la luminosità e la nitidezza delle scene proiettate che sembravano scure e sfuocate e ancora hanno dichiarato di avere dei problemi nel spostare i robot durante le attività (questo dovuto al cambio di velocità quando le batterie dei robot iniziavano a scaricarsi). In generale possiamo riassumere le caratteristiche positive di GENTORO come segue: il fatto che il robot sia come un giocattolo telecomandato lo ha reso molto interessante per i bambini; la possibilità di personalizzare la storia aiutava a sviluppare la creatività e la collaborazione tra di loro. Infine è stato riscontrato come l'uso di GENTORO aumenti il livello di motivazione nei bambini nello svolgere i propri compiti, questo è un elemento chiave nel caso che questo tipo di sistema venga applicato all'insegnamento.[\[21\]](#)



## 2.5 AR for Mathematics and Geometry Education

“Construct3D: A Virtual Reality Application for Mathematics and Geometry Education. Hannes Kaufmann, Dieter Shmalstieg, Michael Wagner 2000”

Construct3D è basato su un sistema di realtà aumentata chiamato “Studierstube”, il quale permette l’interazione e il lavoro di più studenti nello stesso ambiente. L’area di studio in cui è stato usato Construct3D è quello dell’insegnamento della matematica, in particolare della geometria. Uno dei vantaggi principali di questo sistema è la possibilità di visualizzare gli oggetti tridimensionali, mentre prima venivano realizzati su un foglio bidimensionale. Construct3D usa un HMD e un personal interaction panel (PIP), uno strumento di interazione a due mani che prevede l’uso di una penna e un pad, equipaggiati con tracker di posizione e orientamento e un supporto sonoro. L’interfaccia utente è molto semplice: grossi bottoni 3D con icone, anch’esse 3D, che fluttuavano sui bottoni. Questo rende intuitivo e veloce la selezione degli elementi del menù; è possibile selezionare sette diversi elementi base: punto, linea, piano, cubo, sfera, cono e cilindro (gli oggetti solidi sono visualizzati in maniera semitrasparente per permettere di visualizzare anche gli elementi interni). Infine il sistema audio funge da feedback sonoro per dare aiuto e dirigere le azioni dell’utente.

Nel 2000 fu effettuato uno studio pilota per valutare l’utilizzo di Construct3D come strumento per aiutare gli studenti nell’apprendimento dei concetti matematici e di geometria descrittiva. L’esperimento fu effettuato su 14 studenti di Vienna (6 femmine e 8 maschi), con età compresa tra i 22-34 anni. Su 14 partecipanti 13 avevano studiato geometria descrittiva al liceo e 9 erano studenti di geometria con l’obiettivo di diventare professori di liceo. Il vero e proprio studio fu diviso in due fasi: nella prima veniva richiesto ad ogni partecipante di risolvere un problema matematico con l’uso di Construct3D e l’aiuto di un tutor; nella seconda fase ogni partecipante si sottoponeva ad una breve survey che prevedeva domande su Construct3D e sulla realtà virtuale in generale (Figura 2.5).



Figura 2.5: Studente lavora con Construct3D, con un tutor e da solo.

### 2.5.1 Risultati

Tutti i partecipanti impiegavano tra i 6 e i 13 minuti per risolvere il compito. Dalle survey è emerso che gli studenti reputavano stimolante lavorare con la realtà virtuale. Dieci evidenziarono quanto fosse più semplice visualizzare un oggetto 3D in un ambiente d'AR che su un supporto 2D. Solo sei di essi si sentirono un po' intontiti dopo l'utilizzo del HMD e due utenti trovavano che usare uno schermo fosse più semplice che interagire con un sistema d'AR, questo forse deriva dalla difficoltà iniziale dell'uso della penna e del pad con un campo visivo abbastanza piccolo dovuto all'HMD. Inoltre tutti i partecipanti hanno riscontrato problemi di assegnazioni di punti a determinate coordinate, inoltre la coordinazione mano-occhio si è dimostrata essere difficile per individuare un punto in maniera accurata nello spazio 3D senza un feedback aptico. Per questo molti studenti suggerirono di inserire una griglia trasparente con aggancio automatico ai punti. In generale gli studenti hanno apprezzato il menù semplice ed intuitivo, il dispositivo audio e la possibilità di visualizzare e muovere nell'ambiente 3D gli oggetti geometrici. Inoltre un'altra caratteristica molto apprezzata era la possibilità di collaborare nello spazio reale, sia con gli altri che con gli insegnanti.

Nel 2004 fu realizzato uno studio, che coinvolgeva 15 studenti (nove maschi e sei femmine). Questi furono sottoposti, dopo un periodo di training, a interviste e questionari ISO-NORM 9241/10 sull'usabilità. I risultati emersi sono molto importanti: le categorie "suitable for learning" e "suitable for task" ricevettero i punteggi maggiori, inoltre la maggior parte degli studenti pensava che Cosntruct3D fosse facile da usare, efficiente e permettesse un veloce recupero da un errore. Mentre le categorie "self-descriptiveness" e "conformity with user expectations" hanno totalizzato il punteggio minore. Infatti alcuni studenti hanno riscontrato situazioni di disagio nell'uso prolungato del HMD e nell'esperienza dell'AR (cybersickness). [22]

## 2.6 Sistema AR per Modellazione 3D

"3DARModeler: a 3D Modeling System in Augmented Reality Environment. Trien V. Do and Jong-Weon Lee. 2010"

In questo studio viene presentato un sistema di modellazione 3D in ambiente AR chiamato 3DARModeler. Questo sistema combina il metodo tradizionale di interazione con il computer (mouse, tastiera) con un metodo di input tangibile (markers). Lo scopo principale è quello di creare un'interfaccia intuitiva e semplice da usare per aiutare, utenti con poca esperienza nella modellazione 3D, nella creazione di modelli tridimensionali di oggetti semplici (Figura 2.6). L'utente può usare un HMD, ma questo non è indispensabile, per scegliere oggetti base da personalizzare in base alle caratteristiche dell'oggetto e unirli per creare il modello 3D. L'utente può interagire con questo in tre modi diversi: con i markers (interazione tangibile), con la voce (interazione vocale) e con interazione tradizionale (interazione attraverso mouse e tastiera). Questo sistema può essere visto come la versione semplificata di 3D Studio Max in ambiente AR. 3DARModeler usa la libreria

ARToolkit versione 2.71.3 per la realtà aumentata e OpenGL per il render grafico. Esso usa una tecnica di modellazione gerarchica, cioè vengono creati le singole componenti di un determinato oggetto e in seguito vengono modificate in riferimento allo stesso punto di origine.

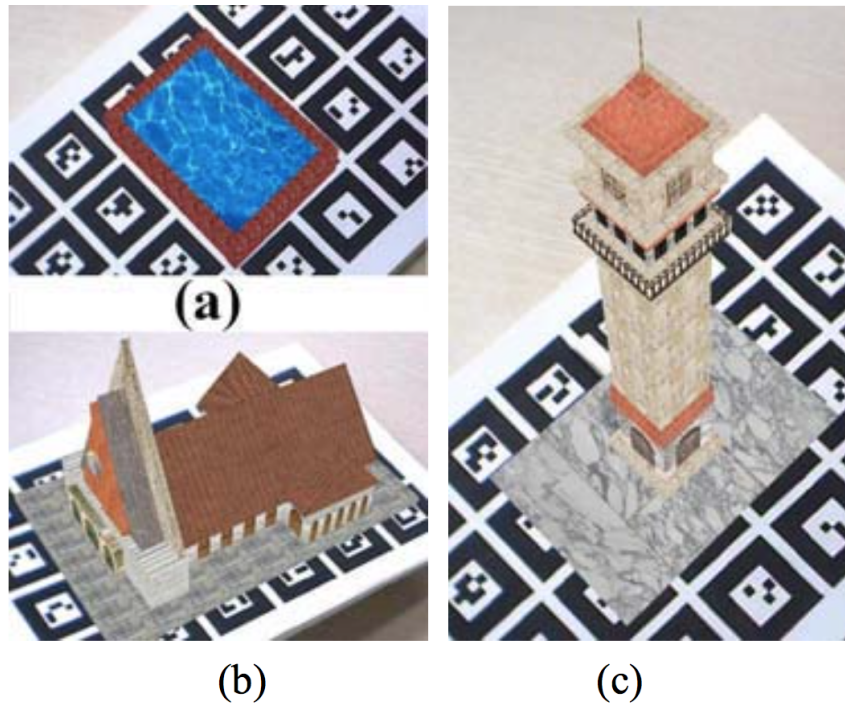


Figura 2.6: Esempi di oggetti creati con 3DARModeler: (a) una piscina, (b) una chiesa, (c) una torre.

### 2.6.1 Studio pilota

Uno studio pilota è stato realizzato per verificare quanto 3DARModeler fosse facile da usare, pratico, comparandolo con 3D Studio Max sotto diversi aspetti. In generale quindi il fine di questo studio era quello di analizzare quanto un sistema AR aiutasse gli utenti nella realizzazione di modelli 3D. Hanno preso parte allo studio trentasette studenti universitari e laureati (dodici femmine e venticinque maschi) tra questi quattordici avevano già avuto esperienza con 3D Studio Max, mentre tutti sapevano niente o poco dell'AR. L'esperimento è stato diviso in tre parti: nella prima parte è stato presentato un video di 10 minuti e spiegato il funzionamento del sistema, subito dopo i partecipanti hanno sperimentato le nozioni apprese usando il sistema per 15 minuti; nella seconda parte agli utenti è stato richiesto di realizzare la nave spaziale mostrata nel video iniziale (le texture presenti furono fornite in anticipo); nella terza ed ultima fase è stato richiesto loro di

rispondere ad un questionario, formato da quindici domande, dieci sul sistema 3DARModeler e cinque sull'ambiente d'AR. Inoltre i partecipanti con esperienza di 3D Studio Max hanno risposto a tre domande in più.

### 2.6.2 Risultati

I risultati hanno riscontrato una generale visione positiva verso il sistema AR. In particolare tutti gli studenti in un tempo relativamente piccolo (tempo medio 17.5 minuti) riuscivano a lavorare senza problemi con il sistema. Tutte le operazioni richieste hanno ricevuto un punteggio alto (4.0 su 5.0) tranne per l'operazione di "selezione degli oggetti" con i markers (3.1 su 5.0); questo era dovuto probabilmente alla differenza di orientamento degli occhi e della telecamera. Complessivamente il sistema è stato definito "interessante" (4.5 su 5.0) e "pratico" (4.0 su 5.0). Ancora è stato richiesto di analizzare le tre modalità di input ed è emerso come la maggior parte degli utenti preferiva la combinazione dei marker e di mouse e tastiera (90%), mentre i comandi vocali incorrevano spesso in errori nel riconoscimento della pronuncia e di conseguenza causava irritazione da parte degli utenti. Le risposte sulla tecnologia d'AR hanno evidenziato come i partecipanti fossero affascinati da questo ambiente, che permetteva di vedere i modelli virtuali come componenti del mondo reale. Infine gli utenti con esperienza 3D Studio Max comparando i due sistemi per la modellazione 3D hanno fatto registrare punteggi molto alti per il sistema che faceva uso della realtà virtuale nelle categorie: "facilità d'uso", "intuitivo", "utile per creare modelli semplici velocemente", mentre punteggio basso per la categoria "utile a creare modelli complessi".

Il vantaggio principale di utilizzare la realtà aumentata in questo sistema di modellazione è che gli utenti sono in grado di creare modelli 3D come se lo stessero facendo nel mondo reale. Il 3DARModeler porta il modo convenzionale e naturale di modellazione 3D alla realtà aumentata. Gli utenti possono creare un modello complesso assemblando geometrie primitive e raggruppandole per formare un oggetto unico. Fornisce un nuovo modo di selezionare, copiare, spostare, incollare e eliminare di oggetti 3D. Con il 3DARModeler, gli utenti sono in grado di progettare rapidamente prototipi, modelli.[\[23\]](#)

## 2.7 AR for Situated Modeling with Tangible Primitive

“Situated modeling: A Shape-Stamping Interface with Tangible Primitives. Manfred Lau, Masaki Hirose, Akira Ohgawara, Jun Mitani, Takeo Igarashi. 2012”

Questo studio analizza un'interfaccia shape-stamping (stampaggio a forma) con la quale gli utenti possono modellare oggetti tridimensionali con primitive 3D tangibili. Quindi gli utenti invece di modellare in uno spazio vuoto, usano queste forme che permettono di utilizzare il mondo reale ed altri oggetti esistenti come guida tangibile durante la modellazione 3D. Questo tipo di interazione prende il nome di modellazione situata: la possibilità di creare nuovi oggetti 3D di dimensioni reali direttamente nello spazio 3D reale usando oggetti esistenti come riferimento (Figura 2.7). L'idea dietro alla modellazione situata è

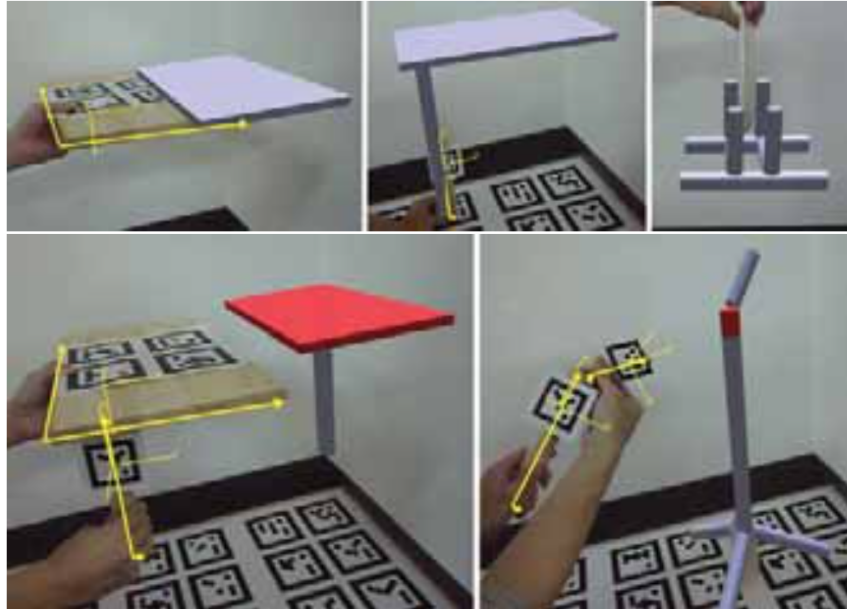


Figura 2.7: Esempio di modellazione situata con uso di primitive tangibili.

quella di incoraggiare gli utenti ad usare direttamente il mondo reale per l'orientamento fisico dell'oggetto virtuale. L'utente può manipolare le forme primitive in uno spazio 3D mettendole direttamente in riferimento con gli oggetti del mondo reale e altre forme base (Figura 2.8). Questa interfaccia è composta molto semplicemente da un HMD e di forme di legno reali sulle quali vengono applicati markers su ogni superficie.

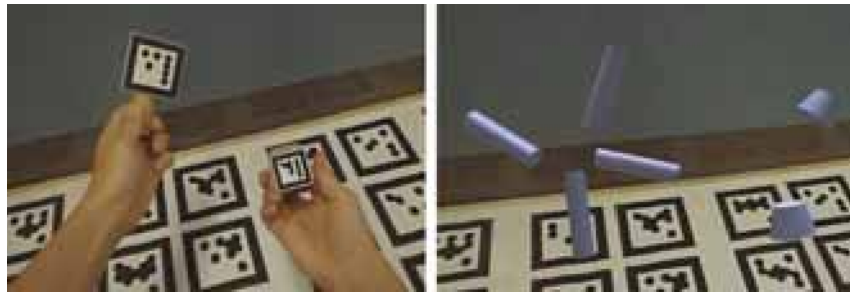


Figura 2.8: Esempio di traslazione e rotazione.

### 2.7.1 Studio

L'obiettivo dello studio è quello di capire quali sono i vantaggi e gli svantaggi dell'interfaccia shape-stamping. Per fare ciò l'interfaccia è stata comparata con un'altra simile già utilizzata denominata Daichi-tools. Quest'ultima permette di costruire e colorare oggetti artistici in tre dimensioni, attraverso l'utilizzo di una serie di dispositivi di input reali basati su strumenti familiari per creare arte in 3D. Questi strumenti diventano metafore "reali", imitando forme e utensili e fornendo sensazioni tattili e audio. In primo luogo è stato realizzato un test pilota con tre utenti che permettesse di definire le caratteristiche migliori e i punti di debolezza, così da consentire di risolvere i problemi riscontrati. Per esempio furono aggiunti markers più grandi sulla parete dell'ambiente poiché il sistema di tracciamento dell'HMD era poco robusto. In seguito è stato realizzato il vero e proprio studio con dieci partecipanti (sei maschi e quattro femmine), studenti universitari, solo tre di loro aveva esperienza di modellazione 3D. I partecipanti sono stati istruiti sul funzionamento delle due interfacce, in seguito è stato dato loro il compito di realizzare una libreria in un ambiente esistente con entrambi i metodi.

### 2.7.2 Risultati

I ricercatori hanno analizzato l'accuratezza dell'uso delle due interfacce registrando cinque misure (altezza, larghezza e profondità dell'intera forma, e le distanze tra piano e i due pizzi orizzontali della libreria) per ogni modello e ogni interfaccia utilizzata per realizzarlo. Confrontando queste misurazioni con le misure reali della libreria data come esempio sono state trovate significative differenze tra le due interfacce. In particolare la shape-stamping permette una precisione superiore a Daichi-tools (le dimensioni delle librerie create con la seconda tendono ad essere più grandi di quelle della prima). Inoltre è stato registrato il tempo di esecuzione delle attività per ogni utente e ogni interfaccia. Il tempo necessario per creare la stessa libreria è minore per la shape-stamping che per Daichi-tools.

Inoltre sono stati richiesti punti di forza e di debolezza di entrambe le interfacce, un punto di forza di shape-stamping è che rende facile immaginare il posizionamento degli oggetti grazie alla dimensione della forma virtuale. Una debolezza invece è il fatto che non riesca a riconoscere i markers quando la forma primitiva è relativamente grande.



Per quanto riguardai Daichi-tools agli utenti era piaciuto l’utensile da taglio, mentre hanno avuto difficoltà nel creare oggetti che toccavano il pavimento o la parete perchè lo strumento “mossa” non è abbastanza preciso rispetto agli oggetti reali che gli utenti tenevano in mano. In generale l’interfaccia shape-stamping ha ricevuto punteggi migliori per le caratteristiche principali: facilità d’uso (4.7 rispetto a 3.1 per Daichi-tools), facile da imparare (6.5 rispetto a 5.9 per Daichi-tools), ottenere controllo preciso (4 rispetto a 2.7 per Daichi-tools) e stanchezza dall’uso (5 rispetto a 6.1 per Daichi-tools).[\[24\]](#)

## 2.8 AR Smartphone Application to Create 3D Models

“Learning to create 3D Models via Augmented Reality Smartphone Interface. Jeff K.T. Tang, Tin-Yung Au Duong, Yui-Wang Ng, Hoi-Kit Luk. 2015”

In questo articolo viene proposta un'app mobile che aiuta la creazione di modelli 3D e a facilitare i concetti della modellazione 3D. Un'applicazione che utilizza l'AR e adotta un approccio “bottom-up” di modellazione: usando diversi markers, che rappresentano forme primitive diverse, e unendole con le proprie mani per formare modelli più grandi, come se si usassero i lego. Infatti le primitive sono come dei mattoncini che vengono visualizzati sui marker stampati e possono essere ruotati, traslati nello stesso modo con cui si manipolano i blocchi di costruzione veri e propri (Figura 2.9).

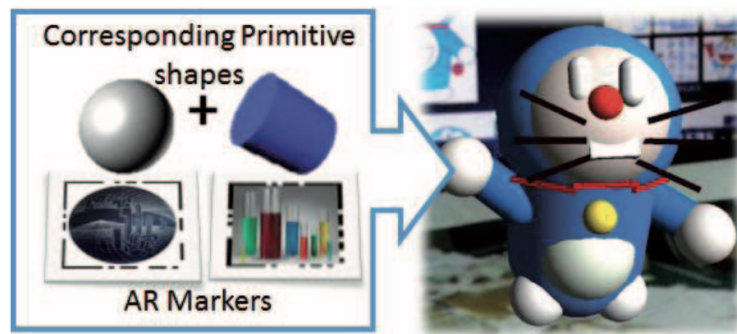


Figura 2.9: Modalità di creazione degli oggetti, partendo da semplici primitive per arrivare all'oggetto finale.



La creazione di un'applicazione mobile invece di una per computer ha portato numerosi vantaggi. Anzitutto lo smartphone è sempre a portata di mano, è come una videocamera portatile e permette agli utenti di creare dei modelli 3D in qualsiasi momento e ovunque. Secondariamente la possibilità di avere uno schermo touch permette una maggiore interattività e quindi favorisce il collegamento tra mondo virtuale e l'utente stesso. Ancora lo smart phone permette un'interazione naturale con gli oggetti 3D, che è simile all'interazione quotidiana con gli oggetti reali (Figura 2.10).

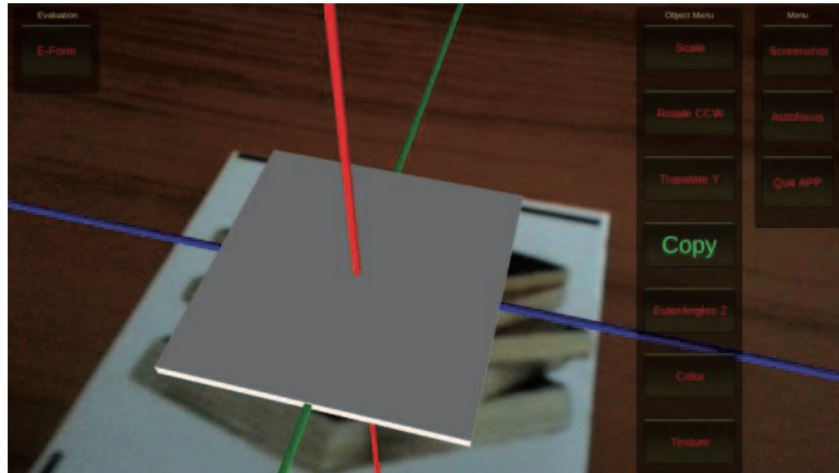


Figura 2.10: Immagine interfaccia utente dell'applicazione.

Per la realizzazione vera e propria dell'applicazione sono state usate la libreria Vuforia per la parte d'AR e quindi del riconoscimento dei markers, e OpenGL e Unity3D per il rendering dei modelli tridimensionali. L'utente attraverso l'interfaccia era in grado di scegliere le primitive che gli servivano per un determinato modello e le operazioni che servivano per manipolarle (rotazione, scala, traslazione, cambio texture ecc).

### 2.8.1 Studio

Uno studio è stato realizzato per analizzare come gli studenti si rapportavano a questa applicazione per creare modelli 3D. I partecipanti all'esperimento erano venti studenti che stavano seguendo corsi di grafica 3D, quindi con esperienza di modellazione. Questi sono stati divisi in due gruppi da dieci, il gruppo sperimentale che usava l'app e il gruppo di controllo che usava Unity3D (scelto per la sua interfaccia intuitiva rispetto agli altri software 3D). È stato richiesto ad entrambi i gruppi di creare il modello 3D di una sedia. Gli sperimentatori hanno calcolato il tempo impiegato da ogni studente e collezionato le risposte ad un piccolo questionario in cui dovevano dare un punteggio massimo di 7 sull'usabilità e l'esperienza in generale.

### 2.8.2 Risultati

I dati raccolti hanno sottolineato come l'uso dell'app fosse facile e naturale, infatti la categoria naturalezza ha avuto un punteggio di 5.6 per l'app e di 5.4 per Unity3D. Anche l'interfaccia del menù molto semplice e le funzioni che apparivano automaticamente, sono state molto apprezzate, ricevendo un punteggio di 5.1 contro il 4.7 dell'esperimento con Unity3D. Per quanto riguarda la facilità d'uso e la stabilità dei software i punteggi dei due gruppi sono molto simili (5.7 e 5.4 rispettivamente). [25]

## 2.9 Conclusioni

Dall'analisi effettuata nei paragrafi precedenti risulta evidente come la realtà aumentata sia una tecnologia molto valida per l'insegnamento. I risultati degli studi hanno rilevato che la maggior parte delle applicazioni di AR per l'educazione sono di tipo marker-based, questo perchè attualmente il processo di tracking è molto più stabile rispetto a quello marker-less. Anche la tecnologia location-based è molto usata soprattutto rispetto a quella marker-less, questo grazie alla presenza di strumenti quali accelerometri, giroscopi, bussola e l'uso del GPS su dispositivi mobile.

Lo scopo ultimo della realtà aumentata, come si evidenzia dalle applicazioni che utilizzano questa tecnologia, è quello di spiegare un argomento e per accrescere le informazioni. L'efficacia della realtà aumentata emerge soprattutto in progetti che hanno come focus l'insegnamento di materie scientifiche quali la matematica, la geometria o la chimica, infatti la possibilità di visualizzare oggetti (ad esempio i solidi o molecole) permette di approcciarsi a problemi complessi con elementi non su un foglio di carta, ma nella "realtà" rendendo compiti complessi molto più semplici. In particolare si può vedere come l'uso di questa tecnologia abbia notevole impatto sulle prestazioni di apprendimento ed induca lo studente ad avere una maggiore motivazione ad imparare e un "atteggiamento positivo" verso l'insegnamento.

Un altro campo dove viene molto utilizzata è l'insegnamento delle lingue e le arti visive poiché permette, da una parte di offrire informazioni "aumentate" combinate con il contesto reale e dall'altra la possibilità di una esperienza visiva migliorata. Le maggiori problematiche emerse, di contro, sono: le limitazioni dovute ad applicazioni mal costruite e l'iniziale confusione che si può avere con una nuova tecnologia. Un esempio è l'utilizzo per la prima volta di device come l'HDM che possono anche essere ingombranti e pesanti.

## Capitolo 3

# Tecnologie Utilizzate

### 3.1 Blender

Blender è un software open source principalmente utilizzato per modellazione, texturing, animazione, rendering. Questo software è stato sviluppato dalla Neo Geo (studio di animazione olandese) come applicazione interna per la Computer Graphics. L'autore principale di Blender è Ton Roosendaal che nel 1998 fondò la “Not a Number Technologies” per sviluppare il programma e distribuirlo. Dopo il fallimento dell'azienda Ton Roosendaal fondò la Blender Foundation, della quale è ancora oggi presidente, per guidare lo sviluppo futuro del software.

Blender consente di realizzare modelli e applicazioni 3D come ad esempio film e videogiochi. La possibilità di importare ed esportare i modelli, aiuta l'interazione con altri programmi e la riusabilità del lavoro svolto, data la grande compatibilità con altre piattaforme. Il sistema di fisica di Blender permette di manipolare elementi simulando anche lo stato liquido o gassoso, gestendo anche collisioni e gravità. Queste caratteristiche lo rendono un programma molto versatile e molto usato data la sua grande flessibilità. La possibilità non solo di modellare, ma di creare animazioni, rende Blender un programma completo.

#### 3.1.1 Principali caratteristiche

L'interfaccia base di Blender (Figura 3.1) è semplice ed essenziale. Le varie funzionalità sono raggruppate sotto diversi menù contestuali che possono essere personalizzati dall'utente. Premendo la barra spaziatrice nella finestra principale è possibile far comparire un menù di ricerca che permette di individuare immediatamente i comandi ricercati. Anche lo schermo è personalizzabile, è infatti possibile dividerlo in più sezioni in modo da avere visuali in modalità differenti o vedere lo stesso oggetto da angolature diverse. Non vi è limite al numero di suddivisioni che si possono fare, anche se tendenzialmente tre o quattro sono più che sufficienti per qualsiasi lavoro. Tutti i comandi possono essere effettuati tramite shortcut da tastiera il che accelera notevolmente la velocità di lavoro.

La schermata iniziale di Blender presenta unicamente un cubo, tuttavia è possibile introdurre altri elementi utilizzando l'apposita funzionalità o tramite la shortcut shift+a.

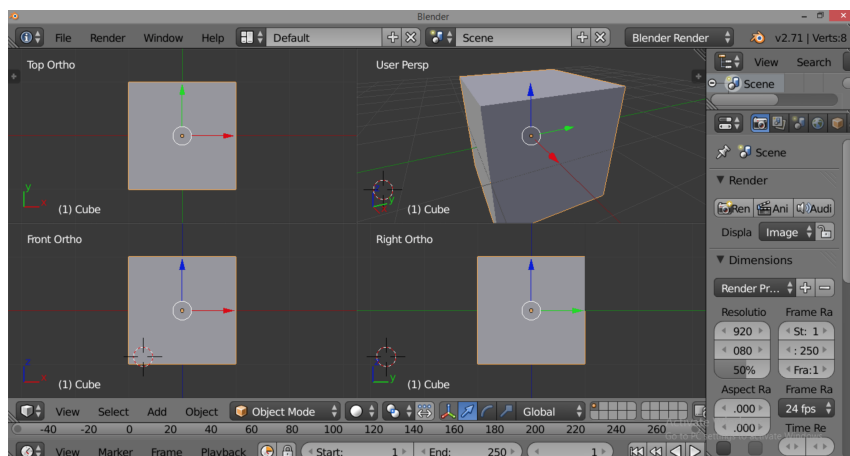


Figura 3.1: Interfaccia principale di Blender.

Questo farà aprire un menù in cui si potrà scegliere il tipo di oggetto da inserire, sia esso una delle primitive proposte dal programma piuttosto che un piano o una curva. In qualsiasi momento è possibile passare dalla selezione di un oggetto all'altro mediante l'uso del tasto destro del mouse.

Le modalità principali di lavoro sono Object mode ed Edit mode. Object mode viene usata per manipolare i singoli oggetti, muovere all'interno dello spazio, scalare o ruotare l'intera mesh. Edit mode è invece utilizzata per modificare gli elementi di un oggetto, permettendo di operare su facce, vertici e spigoli. Per permettere di lavorare in modo più fluido, Blender consente non solo di spostare gli oggetti all'interno della scena, ma anche di muoversi all'interno di essa. Premendo shift e il tasto centrale del mouse è infatti possibile traslare la vista, mentre il solo uso del tasto centrale consente di ruotare la vista. Inoltre l'uso della rotella per incrementare o decrementare lo zoom, aiuta sia ad avere una vista più d'insieme della scena, che ad avvicinarsi agli oggetti per operare in maniera più precisa sui dettagli.

Su Blender è possibile assegnare dei nomi specifici agli oggetti, in modo da identificarli velocemente durante il progetto. Uno dei metodi più usati per realizzare oggetti su Blender, è quello di partire da un oggetto predefinito: un cubo, una sfera o ancora alcuni tipi di curve (NURBS, Bezier). All'interno di Blender è possibile impostare un sistema metrico e visualizzare la lunghezza dei singoli lati o l'ampiezza degli angoli. Altro grande aiuto in questo ambito è la possibilità di impostare un'immagine (blueprint) di background durante il lavoro.

Con il sistema di fisica è possibile impostare diversi parametri per ottenere l'effetto desiderato. Nel caso dei liquidi, ad esempio, è possibile gestire le collisioni e l'erogazione, oltre a specifiche caratteristiche a seconda del fluido che si vuole emulare. Emulare un liquido su un'ampia area risulta tuttavia molto pesante da calcolare e riprodurre. Per questo Blender consente di utilizzare delle forme di controllo (fluid boundary) per evitare che il fluido in questione vada oltre una certa area e influenzi l'ambiente in modo indesiderato.

### 3.1.2 Animazione

Selezionando la modalità animazione nel menù “Screen Layout” l’interfaccia cambia aprendo l’editor, la “Timeline” in basso, il “Dope Sheet”, il “Graph Editor” e la “3D View” standard.

Il “Dope Sheet” è una timeline più evoluta nella quale è possibile gestire i keyframes (duplicarli, spostarli e scalarli) come se si interagisse con gli oggetti nella “3D View”. Per inserire un nuovo keyframe è sufficiente selezionare l’oggetto da animare, nella “3D View”, e premere il tasto “I”, questa operazione apre un menù che permette di selezionare la caratteristica sulla quale inserire il keyframe: (traslazione, rotazione, scalamento ecc...) o su più di una contemporaneamente. Una volta definiti un keyframe all’inizio e alla fine di una singola azione Blender provvede a creare l’animazione completa attraverso un processo di interpolazione, questo non è altro che un calcolo matematico dell’azione e quindi dei frame intermedi del movimento che compie l’oggetto.

Il motore grafico di Blender consente diversi tipi di interpolazione. Queste possono essere visualizzate nel “Graph Editor” che si trova di default sotto il “Dope Sheet”. Blender crea automaticamente un’interpolazione di tipo Bezier, in cui tramite degli Handle è possibile manipolare i singoli keyframe come se fossero delle curve. Anche in questo caso l’uso del Graph Editor utilizza gli stessi comandi di selezione, rotazione e scala della “3D View”, ma solo su due assi. Per cambiare il tipo di interpolazione basta selezionare una singola curva o tutte e premere il tasto “T”, questa operazione apre un menù con tutti i tipi di interpolazione che si possono utilizzare.

Oltre alla modalità “Dope Sheet” è possibile visualizzare gruppi di keyframes come “Action”, attraverso l’ “Action Editor”. Le Action sono le azioni che compie un oggetto. Ogni volta che vengono creati dei Keyframe, queste vengono assegnate all’oggetto su cui vengono create e per ogni oggetto viene creata una Action che prende il nome dell’oggetto. Le Action sono modificabili e soprattutto riassegnabili ad altri oggetti. L’esempio più comune dell’uso delle “Action”, avviene sulle “Armatures” dei personaggi e le “Bones”, che vengono animate in “Pose mode”.

### 3.1.3 Shape Keys

Le “Shape Keys” (chiavi di forma) permettono la memorizzazione di forme relative a partire da una mesh di base. Queste forme possono venire miscelate su di una base percentuale con altre chiavi di forma in modo da ottenere l’effetto desiderato. Questo tipo di modificatori vengono usati principalmente per le espressioni di personaggi 3D, per esempio per il sorriso o il battito degli occhi di una persona. Partendo da una mesh di base, l’utente abilita le chiavi inserendo la chiave di base. Questo si ottiene selezionando l’oggetto mesh e selezionando il pannello “Shapes”. Quindi si preme il pulsante “Add Shape” Key. Creata la chiave di base, è possibile creare forme chiave aggiuntive nello stesso modo. Con la mesh selezionata, si può sia premere il tasto “I” e quindi scegliere Mesh oppure cliccare sul pulsante “Add Shape Key” come descritto in precedenza. Nel pannello “Shapes”, verrà creata una nuova chiave (“Key1”). Una volta create almeno due chiavi è possibile decidere il grado di deformazione andando a variare il valore della dello slider “Value”. [26]

## 3.2 Unity3D

Unity3D è un ambiente di sviluppo per videogiochi, ha un workflow molto semplice, infatti è adatto per tutte le tipologie di applicazioni, di piccole e grandi dimensioni. All'interno di Unity3D possiamo trovare tutti i principali tool necessari per la creazione di un videogioco come: l'animation tool, il gestore di materiali, il tool per la creazione di un terreno, alberi ecc. . . Grazie al supporto di diverse librerie grafiche quali Direct3D, OpenGL e OpenGL ES, Unity3D permette di sviluppare applicazioni cross-platform. È possibile creare applicazioni per Windows, Mac, Linux, Web, iOS, Android, Windows Phone, Wii, Xbox e Playstation. Ovviamente alcune delle licenze sono a pagamento o necessitano della versione PRO. La versione base di Unity3D è completamente free. La versione PRO differisce da quella free per alcune funzioni come la riflessione dell'acqua, ombre dinamiche, il profiler, utile per l'ottimizzazione del codice, gli effetti di postprocess e tante altre funzioni. Tuttavia anche senza l'utilizzo di tutte queste funzioni è possibile creare applicazioni e giochi di qualità.

### 3.2.1 Interfaccia e principali caratteristiche

L'area di lavoro di Unity è suddivisa in finestre (Figura 3.2):

- “Hierarchy” è la finestra che elenca gli oggetti presenti nella scena, grazie ad essa è possibile organizzare gli oggetti e selezionarli. Un oggetto può contenere altri oggetti al suo interno.
- “Scene” è la finestra che contiene la vista sulla scena attualmente in utilizzo, in questa finestra è possibile collocare, spostare, ruotare e scalare i vari oggetti nello spazio creando l'ambiente. Questa finestra è impostata di default sulla vista 3D, ma tramite un bottone in alto indicato con il simbolo 2D è possibile impostare la scena per una visualizzazione 2D nel caso in cui si voglia creare un gioco a due dimensioni.
- “Game” è la finestra che simula il gioco vero e proprio. Tutto quello che viene visualizzato dalla camera viene reso visibile in questa finestra. Durante il testing di una scena si utilizza questa finestra.
- “Inspector” visualizza tutte le proprietà ed i componenti del game object selezionato.
- “Project” è la finestra dove è possibile visualizzare tutte le risorse del nostro progetto e organizzarle a nostro piacimento.
- “Console” è la finestra dove vengono visualizzati tutti i messaggi di sistema o gli eventuali errori/warnings di compilazione.

Ogni progetto creato con Unity deve contenere tutte le risorse di cui l'applicazione ha bisogno, al suo interno possono essere definite nuove scene, scripts e prefabs. Una scena

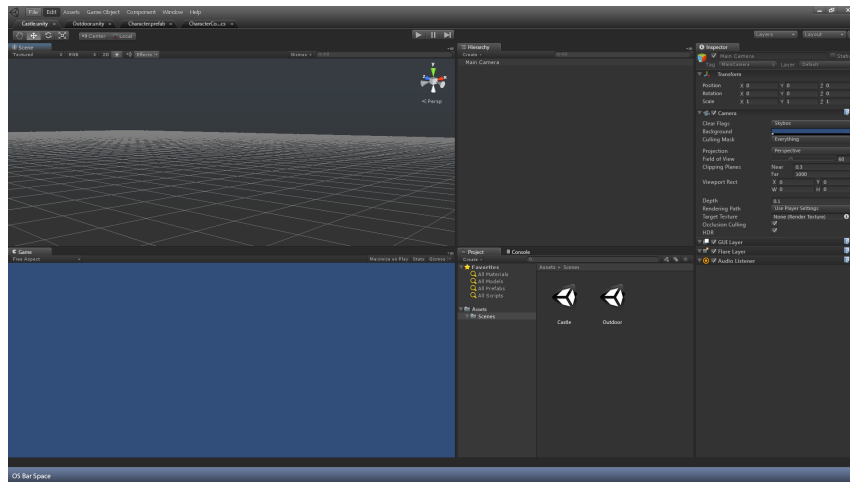


Figura 3.2: Interfaccia principale di Unity3D.

è un ambiente virtuale isolato, in essa sono definiti tutti gli oggetti che compongono un livello. Implicitamente quando viene caricata una scena l'engine alloca tutti gli oggetti in memoria, nel caso in cui si cambia da una scena all'altra tutti gli oggetti della scena precedente vengono deallocati, esistono comunque diverse tecniche per mantenere dei game object tra una scena e l'altra. In Unity prevalgono tre concetti fondamentali, "Game Object", "Component" e "Prefab". Il "Game Object" è la primitiva fondamentale dell'engine a cui si possono aggiungere dei componenti che ne specificano dei comportamenti e delle proprietà. Ogni game object ha un Transform, che definisce la posizione, rotazione e grandezza dell'oggetto secondo un sistema di coordinate xyz. Il "Component" è appunto un componente che va aggiunto ad un game object in base alle esigenze per aggiungere proprietà specifiche all'oggetto o comportamenti che il "Game Object" deve avere allo scatenarsi di un qualche evento. I "Component" più importanti ed utilizzati sono: "Audio Source", "Animation", "Mesh Collider", "Box Collider" e "Scripts". È possibile aggiungere ad un oggetto uno Script semplicemente trascinando lo script dalla finestra project alla finestra Hierarchy sull'oggetto desiderato. Dall'Inspector poi è possibile abilitare, disabilitare o eliminare un Component oppure impostare il valore di una variabile all'interno di uno script.

Ancora Unity permette il salvataggio di oggetti creati nella scena tramite l'utilizzo dei Prefab, elementi generati dalla finestra project che formano un collegamento tra tutti i game object dello stesso tipo. Il vantaggio di utilizzare un prefab risiede non solo nel poter salvare all'interno del progetto i game object più comuni, ma anche nel permettere che una modifica effettuata su di esso si ripercuota su tutte le copie di quell'oggetto, rendendo molto più facile la gestione delle copie.

### 3.2.2 Animation Tool

Un tool fondamentale è senz'altro l'Animation Tool. Questo tool permette di creare animazioni da assegnare poi ai vari game object. Con la parola animazione non si intende solamente la traslazione e la rotazione ma anche qualsiasi cambio di proprietà dell'oggetto: è possibile far cambiare il colore di un oggetto, è possibile cambiare la variabile pubblica di uno script e tanto altro. L'animation tool si presenta con principalmente due fogli di lavoro: il Dope Sheet e Curves. Il Dope sheet è semplicemente un foglio dove creando delle Key è possibile memorizzare dei cambi di proprietà nel tempo. Una volta create delle Key, e quindi dei cambi di proprietà, su Curves si può osservare la curva che è stata generata. In questo foglio di lavoro è possibile modificare l'andamento dell'animazione nel tempo tramite l'utilizzo di strumenti come: smooth, flat e broken.

### 3.2.3 Scripting

Javascript, C# e Boo sono i tre linguaggi con cui è possibile lavorare su Unity. Negli ultimi anni si è andato sempre più verso un abbandono di Javascript e Boo a beneficio dell'uso di C#. La parte di scripting di Unity viene definita su MonoDevelop, un'implementazione open-source del .NET Framework. Nell'installazione di Unity è inclusa una versione proprietaria di MonoDevelop da poter utilizzare per scrivere il codice.

Ogni script per funzionare deve essere utilizzato come component e quindi “assegnato” ad un game object presente nella scena. Molte funzioni all'interno di uno script sono richiamate solo al verificarsi di un particolare evento relativo a quell'oggetto. Quando si va a creare uno script nel progetto Unity crea una classe che deriva da MonoBehaviour. MonoBehaviour comprende tutte le funzioni più importanti e maggiormente utilizzate per la creazione di un videogames/applicazione:

- Update: chiamata prima che un frame proceda nella fase di rendering, al suo interno è spesso specificata la parte di codice che definisce il comportamento dell'oggetto e/o la gestione degli input.
- FixedUpdate: come per Update, ma viene chiamata prima di ogni step dell'engine fisico. Utilizzata per oggetti Rigidbody ossia quelli che sono affetti dalla fisica.
- OnGUI: richiamata più volte per frame, utilizzata per disegnare a schermo l'interfaccia utente e gestire gli eventi relativi alla GUI (Graphics User Interface)
- Start: è chiamata nel frame in cui lo script viene abilitato prima di qualsiasi altro metodo. La funzione Start viene chiamata una sola volta per tutta la durata dello script. Funge da “costruttore” ha il compito di creare le istanze dei vari oggetti.
- OnTriggerEnter: richiamata quando il Collider del game object a cui è “attaccato” lo script entra dentro un Trigger.
- OnTriggerExit: come la precedente ma viene chiamata quando i collider non si intersecano più.



Le variabili al di fuori delle funzioni sono inizializzate all'abilitazione dello script. Ogni variabile resa pubblica è resa visibile nell'inspector ed è possibile assegnare i valori di quest'ultime tramite l'interfaccia di Unity. Per comunicare da uno script ad un altro esistono diversi metodi. Uno di quelli più comuni è quello dell'utilizzo del metodo `GetComponent`. Con il quale è possibile accedere ad un component e quindi accedere a tutte le funzioni e proprietà pubbliche utilizzando appunto il `GetComponent`: un metodo derivato da game object. Ogni qualvolta si crea uno script nel progetto questo diventa un "tipo" identificato dal nome datogli. Basterà quindi semplicemente inserire il nome dello script come parametro al metodo `GetComponent` e ci si potrà accedere. Questo metodo può essere utilizzato per accedere a tutti i tipi di componenti. In Unity esiste una funzione che permette di mandare messaggi ad un determinato game object inviando come parametro una stringa che contiene il nome di una funzione: se questa esiste sarà eseguita. Nel caso in cui siano presenti più scripts con la stessa funzione queste vengono eseguite contemporaneamente. Esistono anche funzioni che permettono di ottenere i riferimenti di gameobject presenti nella scena. `Find` e `FindGameObjectWithTag` sono due funzioni della classe `GameObject` che permettono uno di trovare un gameobject tramite il nome (`Find`) e l'altro tramite il tag (`FindGameObjectWithTag`). [27]

### 3.3 Vuforia

Vuforia è una piattaforma per la realtà aumentata, la più diffusa e usata al mondo. Con il supporto ai principali cellulari, tablet, visori e occhiali, Vuforia ha reso possibili più di 475 milioni di installazioni di app per AR da Apple Store e Google Play. Vuforia Engine, il cuore della piattaforma, usa telecamere e sensori come un occhio digitale all'interno di una app. L'occhio "vede" gli oggetti e le superfici nei quali può essere collocato il contenuto digitale e permette agli sviluppatori di creare le esperienze d'AR usando gli strumenti di sviluppo esistenti. Vuforia è arrivata alla versione 7 (uscita alla fine del 2017), questa include `Model Targets`, una nuova feature che abilita esperienze d'AR con oggetti non riconoscibili usando la tecnologia di visione computerizzata esistente. `Model Targets` riconosce gli oggetti dalla forma, a differenza di quanto fanno i metodi esistenti, che si basano sui modelli visuali dettagliati che normalmente si trovano su carta stampata, imballaggio di prodotti e molti beni di consumo. Con `Model Targets`, il contenuto può essere abbinato a oggetti come automobili, elettrodomestici e macchinari industriali.

Dalla versione 2017.2 di Unity3D, Vuforia è integrata nell'ambiente di sviluppo per videogames, questo rende molto più semplice progettare applicazioni in AR. Infatti ora basta attivare il supporto a "Vuforia Augmented Reality" nel pannello "XR Settings" in Unity "Player Setting". Una volta attivato il supporto nel menù `GameObject` comparirà una voce "Vuforia", dalla quale poter inserire gli object relativi all'AR, i più importanti ed anche più usati sono:

- AR Camera, che permette di inserire nella scena una camera di tipo AR per intercettare i marker.

- Image (ImageTarget), che invece serve appunto per definire un'immagine che servirà da tracker sulla quale è visualizzata l'informazione in 3D.

Per caricare un'immagine come tracker bisogna creare un database di immagini nella parte del sito di Vuforia dedicato agli sviluppatori. Quando viene creato un nuovo database viene creata una chiave univoca (App Licence Key), che deve essere inserita all'interno del pannello Vuforia Configuration; in seguito bisogna scaricare il DB come progetto Unity e importarlo nel progetto. Una volta nel progetto questo deve essere attivato, mettendo la spunta sul nome del database caricato, nel pannello Vuforia Configuration. Una volta attivato, nell'Inspector relativo all'ImageTarget nella parte relativa al "Image Target Behaviour" è possibile scegliere il database e l'immagine desiderata.

### 3.3.1 Vuforia Scripting

Quando vengono creati gli oggetti AR Camera e ImageTarget vengono automaticamente caricati due script: "DefaultInitializationHandlerError" e "DefaultTrackableEventHandler". Il primo non fa altro che intercettare e comunicare eventuali errori relativi all'inizializzazione e il funzionamento dell'AR. Mentre il secondo permette di interagire con i marker nella scena attraverso tre metodi fondamentali:

- OnTrackingFound: permette di far accadere qualcosa quando viene trovato un determinato tracker.
- OnTrackingLost: al contrario permette di far accadere qualcosa quando il tracking viene perso.
- OnTrackableStateChange: permette di intercettare i cambiamenti relativi al tracking e far accadere qualcosa (per esempio al cambio di marker inquadrato).

[\[28\]](#)

## Capitolo 4

# Progettazione e sviluppo dell'applicazione ARhino

### 4.1 Introduzione

Nella fase di sviluppo viene definita e realizzata l'applicazione. Prima di passare a questa fase sono stati fissati gli obiettivi che si volevano raggiungere ed in base a questi è stata elaborata una soluzione possibile. Conclusa la fase di ricerca, infatti, sono stati analizzati affondo i fattori, le potenzialità e i limiti relativi all'uso dell'AR nell'ambito dell'insegnamento emersi dai testi approfonditi. Questa valutazione ha permesso di definire le specifiche di ARhino. L'idea che è risultata dallo studio preliminare è quella di un'applicazione che utilizzasse dei quiz, sotto forma di "gioco" e delle animazioni in AR per rendere più immediati alcuni passaggi della realizzazione di oggetti tridimensionali in Rhinoceros 3D. Lo scopo finale è quello di cercare di semplificare l'apprendimento di nozioni relative alla modellazione 3D che risultano alquanto ostiche per gli studenti.

### 4.2 Gamification

Prima di continuare con la struttura di Arhino definiamo il concetto di Gamification [29] che svolge un ruolo fondamentale come visto nel paragrafo 1.3.

Il termine deriva dalla parola "Game", cioè gioco, nell'accezione del semplice divertimento senza altri fini. La Gamification però non è solo questo: esso è uno strumento molto efficace nel veicolare determinati messaggi e far sì che gli utenti assumano comportamenti attivi, così da raggiungere obiettivi specifici. In questo caso specifico viene utilizzato per cercare di rendere l'apprendimento meno passivo e riuscire ad innescare un atteggiamento positivo verso lo studio della modellazione 3D. Con l'espansione del mercato videludico questo concetto è diventato sempre più presente e importante nelle strategie di marketing legate a questo mondo.

Si può definire la Gamification come un insieme di regole derivate dal mondo dei videogames che hanno l'obiettivo di applicare meccaniche ludiche ad attività che non hanno direttamente a che fare con il gioco; così facendo è possibile influenzare e modificare il

comportamento degli individui, favorendo la nascita ed il consolidamento di interesse attivo da parte degli utenti coinvolti.

Ci sono molti contesti nei quali è possibile applicare quello che possiamo definire come il “metodo” Gamification: un sito, un servizio, una comunità, un contenuto o campagna sono tutti contesti che possono essere “gamificati” (da “to gamify”) così da far nascere l'interesse, il coinvolgimento e la partecipazione degli utenti. Per raggiungere questi obiettivi, il processo di comunicazione deve necessariamente essere ripensato in modo da introdurre meccaniche e dinamiche di gioco, aggiungendo ai fattori tradizionali altre componenti trainanti (ancora, mutate dal mondo del “gaming”) che possano attirare l'interesse dell'utenza, spingendola a rielaborare specifici contenuti proposti volontariamente e più volte nell'arco del tempo.

Le meccaniche e le dinamiche di gioco rappresentano i veri punti cardine nell'ambito della Gamification, necessari per “gamificare” un'applicazione o un servizio: l'introduzione di concetti come punti, livelli, missioni e sfide. Questo incoraggia gli utenti ad investire il proprio tempo, spingendoli alla partecipazione e aiutandoli a costruire delle relazioni all'interno del gioco. Tali relazioni motivano gli utenti al raggiungimento di obiettivi pre-determinati (es: il miglioramento delle proprie capacità, l'incremento delle performance, ecc), modificando di fatto il loro comportamento.

## 4.3 Struttura ARhino

Nei paragrafi seguenti verrà descritta la struttura dell'applicazione ARhino. Verranno inoltre delineate le scelte fatte e verranno analizzate le motivazioni di tali scelte.

### 4.3.1 Flusso di Utilizzo

Cliccando sull'icona dell'app viene aperta una schermata di caricamento dove vengono visualizzate delle informazioni sugli esercizi e le animazioni, che spiegano le modalità di utilizzo generali dell'applicazioni(Figura 4.1).

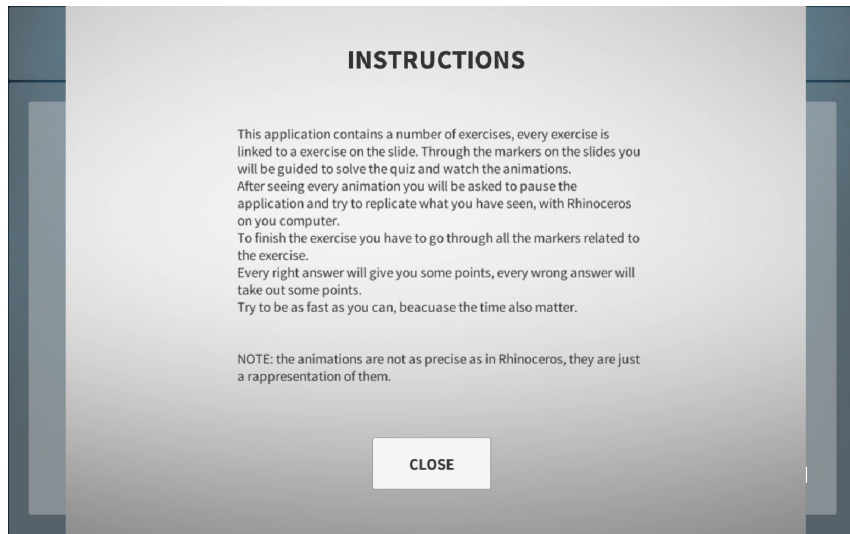


Figura 4.1: Schermata di introduzione dove vengono spiegate le istruzioni.

Dopo che l'utente ha schiacciato il bottone "CLOSE" si apre la schermata principale: il menù (Figura 4.2). In questa schermata sono presenti un numero di bottoni quanti sono gli esercizi presenti. Al primo utilizzo dell'app soltanto il primo bottone è cliccabile, questo perchè l'idea è quella di far sbloccare gli esercizi successivi come una sorta di ricompensa per aver completato quello precedente (come nei videogames con i "livelli"). All'interno del bottone è possibile visualizzare il tempo impiegato a completare l'esercizio e il punteggio realizzato, quest'ultimo sotto forma di stelline.

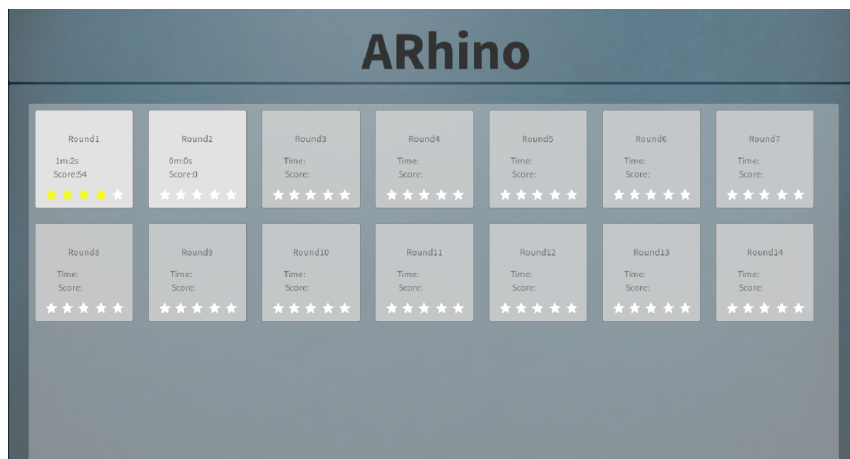


Figura 4.2: Schermata principale ARhino: il menu.

Cliccando sul bottone di un determinato esercizio si aprirà una nuova schermata relativa. In particolare all'inizio di ogni esercizio verrà attivata la camera per inquadrare il primo marker. Quando l'utente inquadra il primo marker verrà caricata sempre la prima domanda relativa a quell'esercizio e verrà fatto partire il tempo. Nella schermata dell'esercizio oltre al pannello delle domande, che occupa la parte centrale dell'interfaccia, abbiamo anche un pannello di tipo testuale che ha la funzione di notificare e dare suggerimenti per il prosieguo (posizionato in basso sotto le domande) e uno relativo allo stato dell'esercizio che comprende il tempo reale, un pannello che notifica l'esattezza o meno delle risposte e un bottone di pausa.

Se la risposta dell'utente è corretta viene visualizzata la domanda successiva e viene assegnato un determinato punteggio che si aggiunge al punteggio complessivo. Mentre se la risposta è sbagliata viene ricaricata la stessa domanda e il punteggio complessivo viene ridotto di un certo valore. La modalità scelta per notificare l'esattezza o meno della risposta è l'utilizzo di una emoji: una faccina verde sorridente se la risposta è corretta e una faccina rossa triste in caso contrario.

Dopo un certo numero di domande, che varia in base all'argomento e alla sua difficoltà, viene richiesto di inquadrare il prossimo marker sul quale appare un'animazione in AR relativa alle domande appena terminate. Al termine di tutte le animazioni verrà chiesto all'utente di mettere in pausa l'applicazione e di provare a realizzare i passaggi appena spiegati direttamente sul computer con il software Rhinoceros 3D. Durante questa fase (Figura 4.3) l'applicazione non sarà utilizzabile, però il tempo continuerà ad andare avanti. Questo meccanismo permette di avere un indice di quanto l'utente abbia chiaro il passaggio appena visualizzato, inoltre il tempo è impiegato per calcolare il punteggio finale. In particolare per ogni esercizio vengono definiti due valori di tempo: se l'utente riesce a realizzare tutto l'esercizio (domande più passaggio sul computer) in un tempo minore del primo valore allora il punteggio non verrà modificato, mentre se il tempo è compreso tra i due valori o maggiore del secondo il punteggio subirà una riduzione di una determinata percentuale.

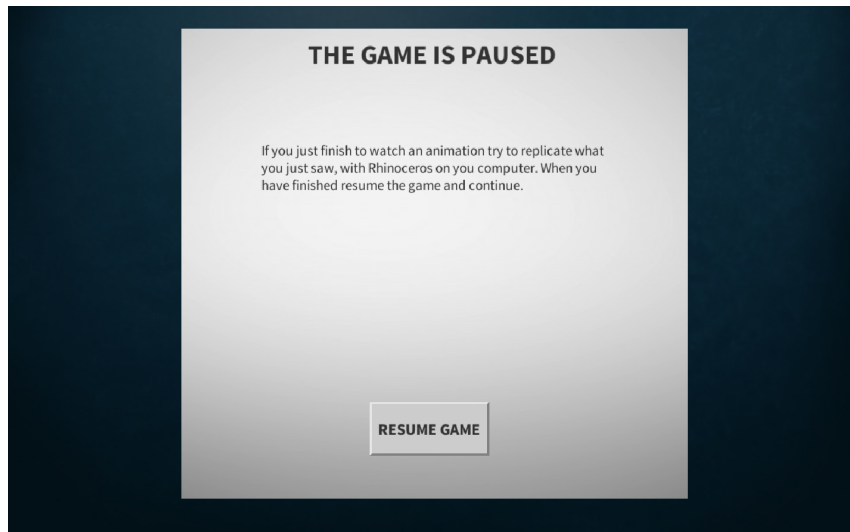


Figura 4.3: Schermata pausa.

Nella schermata degli esercizi si susseguono un numero di domande variabile e animazioni relative ad esse. Ogni esercizio termina con la stessa domanda, che ha lo scopo di capire se lo studente ha chiari tutti i passaggi svolti nell'esercizio. Concluso l'esercizio si apre la schermata del punteggio dove verranno visualizzati il tempo, il punteggio in stelline (da 1 a 5) e un immagine che cambia a seconda del punteggio e del tempo impiegato. In questo caso si è deciso di usare un'immagine "ludica di un cane che esprime tre diversi sentimenti in base al punteggio realizzato: felice, normale e triste (Figura 4.4). Infine cliccando un bottone si torna alla schermata del menù principale, nel quale verrà aggiornato il punteggio nel relativo bottone e verrà sbloccato il bottone dell'esercizio successivo.

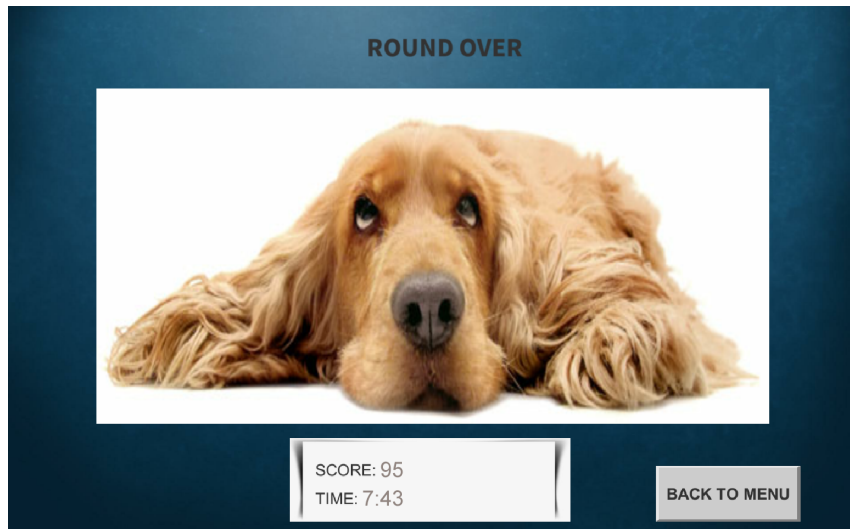


Figura 4.4: Schermata fine esercizio.

#### 4.3.2 Requisiti base

Di seguito verranno elencati i requisiti base dell'applicazione, che sono emersi come fondamentali per il raggiungimento dell'obiettivo prefissato. Il primo e forse anche il più importante è la “possibilità di riprovare”, ad ogni domanda l'utente può rispondere in maniera errata senza però perdere la possibilità di riprovare e rispondere in maniera esatta. In questo modo l'utente non è scoraggiato a continuare e non si crea in lui un atteggiamento negativo verso il quiz.

La seconda è la possibilità di avere un “feedback immediato”, infatti per ogni risposta giusta viene visualizzata un'immagine di una faccina sorridente di colore verde, mentre per ogni risposta sbagliata ne viene visualizzata una triste di colore rosso. Ancora è presente un piccolo pannello nel quale vengono visualizzati di volta in volta suggerimenti e o notifiche in tempo reale, che permettono di capire come andare avanti dell'esercizio o di spronare ad andare avanti.

L'applicazione è stata pensata come un gioco a livelli di difficoltà sempre maggiori, questi seguono gli argomenti contenuti in alcune slides. La struttura del software è divisa in un numero di esercizi, che possono essere realizzati in maniera sequenziale; appena viene concluso il primo viene sbloccato il successivo, come i livelli di un videogame.

Un'altra feature è la possibilità di mettere pausa all'applicazione e realizzare il passo/esercizio sul computer direttamente in Rhinoceros3D. Questo permette un immediato riscontro con quello appena osservato nell'app: l'utente collega al meglio una cosa di cui ha avuto appena esperienza. Infine è stato pensato un “sistema di punteggio” relativo ad ogni esercizio sotto forma di stelline, questo punteggio è calcolato in base alle risposte esatte e sbagliate e il tempo impiegato.



### 4.3.3 Le Animazioni

Si è deciso di realizzare un numero limitato di animazioni in base a quelle relative ai passaggi più complessi. Queste animazioni sono state realizzate utilizzando il software Blender, realizzando sia gli oggetti 3D che le animazioni. In un secondo momento queste sono state trasferite in Unity 3D utilizzando il formato FBX. Le animazioni create sono il più fedeli possibile alle funzioni di Rhinoceros 3D rappresentate, anche se con precisione inferiore. Un esempio di differenza tra i l'applicazione Rhino 3D e le animazioni in Blender è l'impossibilità di eliminare alcuni oggetti utilizzati per determinati passaggi. Il modo usato per risolvere questo problema è stato di muovere determinati oggetti il più lontano possibile dall'inquadratura.

## 4.4 Architettura ARhino

L'obiettivo finale, come accennato nel paragrafo 1.5, del progetto è quello di realizzare un'applicazione per gli studenti dell'esame di "Design di Scenario" del corso di laurea in Design e Comunicazione Visiva del Politecnico di Torino in grado di fornire un supporto all'insegnamento della modellazione 3D con l'utilizzo della realtà aumentata. Di seguito viene presentata l'architettura (Figura 4.1) che si è scelto di utilizzare per realizzare tale progetto.

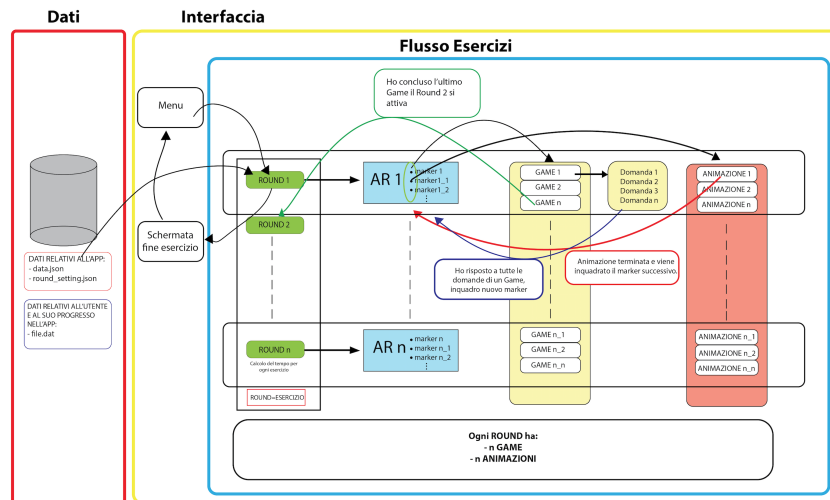


Figura 4.5: Architettura Software ARhino.

Il pattern usato è il Model-View-Controller (MVC), il più usato nella programmazione object oriented e nello sviluppo di interfacce grafiche, poiché rappresenta uno dei concetti fondamentali della programmazione ad oggetti e permette di strutturare l'applicazione in maniera molto efficiente. Infatti, ogni classe viene progettata in modo da svolgere compiti specifici. Inoltre questi rappresentano una classificazione dell'oggetto che si utilizza e risultano essere elementi logicamente separati, che però sono tra di loro in stretta comunicazione. Il grafico in (Figura 4.6) illustra lo schema del design pattern utilizzato.

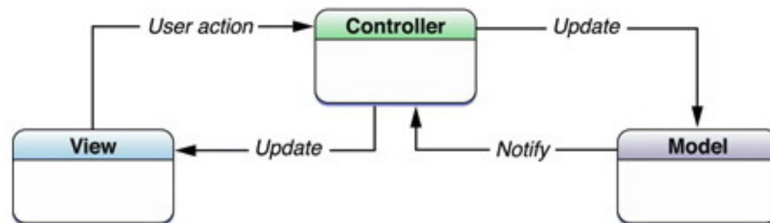


Figura 4.6: Schema pattern MVC.

Il pattern non stabilisce soltanto il ruolo che un determinato oggetto/classe deve avere all'interno dell'applicazione, ma anche il modo in cui gli oggetti comunicano tra loro. Analizzando la (Figura 4.6) è possibile approfondire i tipi di oggetti che il pattern definisce:

- **Model:** contiene i dati dell'applicativo e si occupa di definire le varie procedure che vengono effettuate su di essi, sia in lettura che in scrittura. Il modello non può avere connessione diretta con un oggetto di tipo view, poiché ha il compito di gestire i dati che non devono essere legati ad un particolare tipo di visualizzazione.
- **View:** il ruolo della vista è quello di presentare all'utente i dati contenuti all'interno di un modello. Concettualmente il modello è un oggetto non concreto, mentre la vista lo è, in quanto l'utente può interagire con essa. La vista è dunque una realizzazione di un oggetto non concreto e mette a disposizione un'interfaccia per la modifica dei dati contenuti nel modello. L'oggetto di tipo vista non deve avere un riferimento esplicito ad un oggetto di tipo modello quindi a questo punto viene usato l'oggetto controllore che si frappone tra gli altri due oggetti.
- **Controller:** questo svolge la funzione di intermediario tra oggetti di tipo vista ed oggetti di tipo modello. Un singolo controllore può avere un numero di relazioni arbitrarie tra oggetti di tipo modello e vista, che possono essere relazioni uno a uno

o molti a molti. Il controllore si occupa di inizializzare la vista con i dati contenuti nel modello e informare la vista delle modifiche sui dati subite dall'interazione dell'utente.

#### 4.4.1 View ARhino

L'interfaccia utente è stata progettata seguendo un semplice layout che rende intuitivo l'utilizzo. Nella schermata principale, quella del menù, è stato utilizzato il GridLayout come impaginazione della vista, questo ha reso possibile disporre i bottoni relativi agli esercizi in una griglia che viene popolata da sinistra a destra e dall'alto verso il basso a seconda del numero di esercizi presenti. In Unity 3D il GridLayout è molto semplice da utilizzare poiché può essere scelto come Component da collegare al canvas in cui sono presenti i bottoni. Per la realizzazione dei bottoni e di tutta la parte relativa all'interfaccia grafica di ARhino sono stati utilizzati i componenti UI(user interface) base già presenti in Unity 3D.

La schermata relativa all'esercizio è formata da diversi canvas di cui il principale è quello centrale. Questo permette di visualizzare le domande con le risposte possibili, utilizzando un processo in run time che permette una scalabilità immediata dell'app a seconda dei dati inseriti, questa è la stessa operazione utilizzata per la visualizzazione dei bottoni del menù (questi processi verranno approfonditi nei prossimi paragrafi). Il canvas superiore permette visualizzati dati relativi all'esercizio quali tempo, punteggio in tempo reale e feedback sulle risposte. In particolare nel canvas superiore è presente un pulsante di pausa, questo una volta cliccato permette di stoppare il flusso delle domande e delle animazioni per permettere di realizzare quello che si è appena visto direttamente sul computer. Infine il canvas inferiore che ha lo scopo di comunicare con l'utente.

L'ultima schermata quella di fine esercizio è molto simile a quella precedente con un canvas principale centrale dove vengono visualizzati un'immagine, il tempo finale e il punteggio.

#### 4.4.2 Model ARhino

I dati contenuti nell'applicazione sono salvati in due file di tipo JSON chiamati "data" e "round\_settings". Il JSON, acronimo di Javascript Object Notation, è un formato di testo concepito per l'interscambio di dati all'interno di applicativi di tipo client-server. Questo formato, tuttavia è largamente utilizzato in associazione a linguaggi di programmazione come Java, C, C# e Perl.

Il file "data.json" contiene tutti i dati relativi agli esercizi, come ad esempio le domande e le risposte degli esercizi, il punteggio assegnato per ogni risposta corretta e quello detratto per ogni risposta sbagliata o ancora i valori di tempo per il calcolo finale del punteggio. Il file "round\_settings" invece include i dati dei "game" (dove per game si definisce insieme di domande e animazioni di un determinato argomento) e dei markers di ogni esercizio. La scelta di usare il JSON come formato permette di avere un'applicazione scalabile, che dà la possibilità di apportare modifiche all'applicazione in maniera semplice e veloce, come ad esempio aumentare il numero di esercizi, modificare il numero di risposte ad una o più domande o ancora modificare il numero di markers presenti in ogni esercizio.

Per collegarsi con il file JSON e utilizzare i dati sono state create in Unity 3D delle classi che definiscono il tipo di dato e come questi sono salvati. Un esempio di classe è la classe “Level.cs”.

---

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [System.Serializable]
6 public class Level {
7
8     public int indexLevel;
9     public string Name;
10    public int Unlocked;
11    public bool IsInteractable;
12 }
13
14 }
```

---

Elenco dei codici 4.1: Classe Level.cs

Alcune di queste classi (vedi codice 4.1), che contengono i dati e fungono da modello dell'app sono definite come “Serializable”, questo permette di inserire i dati direttamente dall'inspector del software utilizzato. In Unity 3D inoltre è facile modificare i file JSON grazie alla possibilità di creare un editor nel programma stesso che attraverso una semplice interfaccia consente di aggiungere o cancellare i dati salvati direttamente sul file JSON salvato nel progetto. Come possiamo notare dal codice ?? questo script contiene due metodi fondamentali LoadGameData() che permette di caricare i dati già inseriti e SaveGameData() che permette di salvare i dati una volta modificati nell'Editor. Un altro aspetto da evidenziare sono i due differenti percorsi indicati all'inizio: il primo “/StreamingAssets/data.json” è utilizzato quando viene testata l'app sul computer e il secondo “/storage/emulated/0/DCIM/data.json” viene utilizzato quando l'app è installata sul dispositivo Android.

#### 4.4.3 Controller ARhino

Le classi più importanti come in ogni applicazione che si avvale del pattern MVC sono i controllori. Nel caso dell'applicazione ARhino vengono utilizzati tre controller: il DataController, il GameController e il MenuScreenController.

##### DataController

Il DataController è quello che gestisce i dati al più basso livello. Il suo compito principale è quello di caricare e salvare i dati sia tra una scena e un'altra sia tra una sessione e un'altra. In Unity 3D ogni schermata è definita “scena” e i dati di una scena non vengono automaticamente salvati tra diverse scene. Tuttavia questo è possibile grazie alla scena Persistent che ha proprio lo scopo di richiamare il metodo DontDestroyOnLoad sul

DataController. Questo metodo fa sì che un determinato GameObject, in questo caso il DataController, non venga distrutto automaticamente al passaggio del flusso dell'app da una scena all'altra. Questa scena viene chiamata ogni volta che viene aperta l'applicazione e rimane attiva fino alla chiusura della stessa.

Come detto prima i dati vengono salvati anche tra una sessione e un'altra, questo permette di avere i dati relativi agli esercizi già svolti in sessioni precedenti. Per fare ciò sono stati utilizzati due metodi che caricano e salvano questi dati in un file di testo "save.dat".

---

```
1
2 public void Load()
3     {
4         string destination = "";
5         if(File.Exists (Application.persistentDataPath +
6             "/save.dat"))
7             destination =
8                 Application.persistentDataPath +
9                 "/save.dat";
10        else
11            destination =
12                "/storage/emulated/0/DCIM/save.dat";
13        //string destination =
14            Application.persistentDataPath + "/save.dat";
15        FileStream file;
16
17        if(File.Exists(destination)) file =
18            File.OpenRead(destination);
19        else
20        {
21            Debug.LogError("File not found");
22            return;
23        }
24
25        BinaryFormatter bf = new BinaryFormatter();
26        PlayerData data = (PlayerData)
27            bf.Deserialize(file);
28        file.Close();
29
30        totalTimeSaved = data.totalTime;
31        currentRoundSaved = data.currentRound;
32
33        current_round = currentRoundSaved;
34        timeOverExerciseArr = data.totalTimeExercise;
35        scoreExerciseArr = data.totalScoreExercise;
```

```
31         if (timeOverExerciseArr.Length == 0 ||
32             timeOverExerciseArr == null ||
33             scoreExerciseArr.Length == 0 ||
34             scoreExerciseArr == null)
35             return;
36     }
```

---

Elenco dei codici 4.2: Metodo per il caricamento dati tra sessioni differenti.

---

```
1
2 public void Save()
3     {
4
5
6         string destination = "";
7         if(File.Exists (Application.persistentDataPath +
8             "/save.dat"))
9             destination =
10                 Application.persistentDataPath +
11                 "/save.dat";
12         else
13             destination =
14                 "/storage/emulated/0/DCIM/save.dat";
15
16         //string destination =
17             Application.persistentDataPath + "/save.dat";
18         FileStream file;
19
20         if(File.Exists(destination)) file =
21             File.OpenWrite(destination);
22         else file = File.Create(destination);
23         int val = getCurrentRound();
24         PlayerData data = new PlayerData(timeOverExercise,
25             val, timeOverExerciseArr, scoreExerciseArr);
26         BinaryFormatter bf = new BinaryFormatter();
27         bf.Serialize(file, data);
28         file.Close();
29     }
```

---

Elenco dei codici 4.3: Metodo per il salvataggio dati tra sessioni differenti.

Anche in due casi come per la classe `GameDataEditor` (??) possiamo notare due percorsi diversi per il salvataggio e caricamento dei dati a seconda che si stia lavorando sul pc o sul dispositivo mobile: rispettivamente `"/storage/emulated/0/DCIM/save.dat"` e `"/storage/emulated/0/DCIM/save.dat"`.

## GameController

Il GameController si occupa del flusso degli esercizi, quindi le domande e le relative opzioni di risposta, le scene da far vedere, il calcolo del tempo e del punteggio finale.

Il compito principale di questo controllore è quello di far visualizzare la scena attiva. Nel metodo Start(), che il primo che viene chiamato di una determinata classe, viene inviato attraverso un indice il numero della scena attiva da visualizzare. Questo indice è il numero con il quale finisce il nome di tutti i “game” salvati in Unity come è possibile vedere nell’immagine seguente.

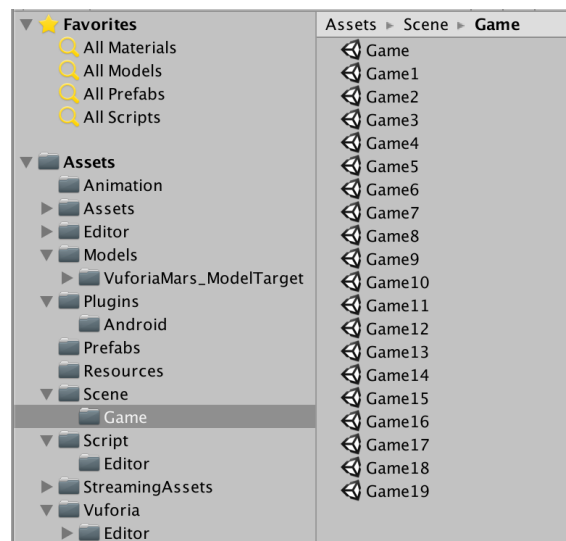


Figura 4.7: Convenzione nomi delle scene di tipo game.

Per il caricamento delle domande è stato utilizzato un sistema particolare di Unity 3D definito come ObjectPool. Questo permette di riutilizzare degli oggetti che vengono usati molte volte durante l'esecuzione del software creato. Nel caso specifico l'oggetto che viene riutilizzato più volte è il bottone delle risposte. Per fare ciò in Unity 3D viene creato una volta e poi trasformato in un "Prefabs". Un Prefabs è un Game Object pre configurato che viene creato nella scena e salvato nel progetto che può essere istanziato o clonato durante il gioco, in run time.

---

```
1 private void ShowQuestion ()
2     {
3         RemoveAnswerButtons ();
4         QuestionData questionData = questionPool
5             [questionIndex];
6         questionDisplayText.text =
7             questionData.questionText;
8
9         communicationText.text =
10            currentRoundData.communication[questionIndex];
11
12         if(communicationText.text.IndexOf("Stop")!=
13            -1||communicationText.text.IndexOf("Ferma")!=-1){
14             if (countError == 0) {
15                 stop = true;
16             } else {
17                 communicationText.text = "Now
18                    answer correctly!";
19             }
20         }
21
22         for (int i = 0; i < questionData.answers.Length;
23            i++)
24         {
25             GameObject answerButtonGameObject =
26                 answerButtonObjectPool.GetObject ();
27             answerButtonGameObjects.Add
28                 (answerButtonGameObject);
29             answerButtonGameObject.transform.SetParent
30                 (answerButtonParent);
31
32             AnswerButton answerButton =
33                 answerButtonGameObject.GetComponent<AnswerButton>
34                 ();
35             answerButton.Setup (questionData.answers [i]);
36         }
37     }
```



---

Elenco dei codici 4.4: Metodo per la visualizzazione delle domande e relative risposte.

Come possiamo vedere nel metodo 4.4, ogni volta che questo viene chiamato vengono rimosse le domande precedenti con il metodo `RemoveAnswerButton()` e viene richiamato il prefabs, in questo caso `AnswerButton`, per popolare le risposte relative alla domanda successiva: “`AnswerButton answerButton = answerButtonGameObject.GetComponent<AnswerButton>()`”.

Un altro metodo importante per questo controller è il metodo che calcola il punteggio. Come già accennato precedentemente il punteggio viene calcolato sommando i punti relativi ad ogni risposta esatta o sottraendo quelli per ogni risposta errata. Alla fine di ogni esercizio questo punteggio subisce una decurtazione di un 5% o 10% a seconda dei valori di soglia del tempo stabiliti. Come possiamo vedere nel 4.5.

---

```
1
2 public int calculateScore(int current_score, float time, int t2,
   int t1){
3
4         int ret = current_score;
5         if (time > t2)
6         {
7                 ret = current_score -((current_score*10) /
10                100);
8         }else if (time > t1) {
9                 ret = current_score -((current_score*5) /
11                100);
12         }
        return ret;
}
```

---

Elenco dei codici 4.5: Metodo per il calcolo finale del punteggio tenendo conto dei limiti di tempo.

### MenuScreenController

Le funzioni principali di quest'ultimo controllore sono: creare la griglia da visualizzare con i bottoni relativi a tutti gli esercizi presenti e gestire i dati presenti in essi. Come già spiegato nel paragrafo `GameController`, anche in questo caso il pulsante relativo ad un esercizio viene creato una sola volta e ne viene creato il Prefabs così da poter essere clonato tante volte quante sono gli esercizi. Tutte le funzionalità più importanti di questo controller sono realizzate dal metodo `FillList()`. In seguito vedremo in particolare le parti del codice che servono per eseguire le varie funzioni.

```
1
2 foreach (var level in LevelList) {
3
4     GameObject newbutton = Instantiate (levelButton) as
        GameObject;
5     LevelButton button = newbutton.GetComponent<LevelButton>
        ();
6     button.LevelText.text = level.Name;
7
8     if (level.indexLevel - 1 <=
        __dataController.getCurrentRound ()) {
9         TimeSpan t = TimeSpan.FromSeconds
            (Mathf.Round
10             (__dataController.timeOverExerciseArr
                [level.indexLevel - 1]));
11
12
13         button.TimeExercise.text =
            t.Minutes+"m:"+t.Seconds+"s";
14     }
15
16     if (level.indexLevel - 1 <=
        __dataController.getCurrentRound ()) {
17         button.ScoreText.text = "Score:" +
            __dataController.scoreExerciseArr [level.indexLevel -
18             1].ToString ();
        }
```

---

Elenco dei codici 4.6: Pezzo del metodo FillList() che permette di istanziare i bottoni relativi agli esercizi .

Come possiamo vedere nel 4.6 per ogni valore di level nella lista LevelList viene istanziato un LevelButton, che è il Prefabs e vengono inseriti il tempo e il punteggio relativo a quell'esercizio (se l'esercizio non è stato ancora eseguito il tempo e il punteggio sono uguali a zero).

---

```
1 var time = __dataController.getTime ();
2
3     if (time > 0 && level.indexLevel <=
        __dataController.getCurrentRound ()) {
4
5     if (__dataController.retriveScoreByIndex (level.indexLevel
        - 1) == 0) {
6
7         button.Star1.SetActive (false);
8         button.Star2.SetActive (false);
9         button.Star3.SetActive (false);
```

```

10         button.Star4.SetActive (false);
11         button.Star5.SetActive (false);
12
13
14     } else if (__dataController.retrieveScoreByIndex
15         (level.indexLevel - 1) > 0 &&
16         __dataController.retrieveScoreByIndex (level.indexLevel
17             - 1) < 10) {
18
19         button.Star1.SetActive (true);
20         button.Star2.SetActive (false);
21         button.Star3.SetActive (false);
22         button.Star4.SetActive (false);
23         button.Star5.SetActive (false);
24
25     } else if (__dataController.retrieveScoreByIndex
26         (level.indexLevel - 1) >= 10 &&
27         __dataController.retrieveScoreByIndex (level.indexLevel
28             - 1) < 30) {
29
30         button.Star1.SetActive (true);
31         button.Star2.SetActive (true);
32         button.Star3.SetActive (false);
33         button.Star4.SetActive (false);
34         button.Star5.SetActive (false);
35
36     } else if (__dataController.retrieveScoreByIndex
37         (level.indexLevel - 1) >= 30 &&
38         __dataController.retrieveScoreByIndex (level.indexLevel
39             - 1) < 50) {
40
41         button.Star1.SetActive (true);
42         button.Star2.SetActive (true);
43         button.Star3.SetActive (true);
44         button.Star4.SetActive (false);
45         button.Star5.SetActive (false);
46
47     } else if (__dataController.retrieveScoreByIndex
48         (level.indexLevel - 1) >= 51 &&
49         __dataController.retrieveScoreByIndex (level.indexLevel
50             - 1) < 70) {
51
52         button.Star1.SetActive (true);
53         button.Star2.SetActive (true);
54         button.Star3.SetActive (true);
55         button.Star4.SetActive (true);
56     }

```

```
47         button.Star5.SetActive (false);
48
49
50     } else if (__dataController.retrieveScoreByIndex
51               (level.indexLevel - 1) > 70) {
52
53         button.Star1.SetActive (true);
54         button.Star2.SetActive (true);
55         button.Star3.SetActive (true);
56         button.Star4.SetActive (true);
57         button.Star5.SetActive (true);
58     }
59 }
```

---

Elenco dei codici 4.7: Pezzo del metodo FillList() che regola l'accensione delle stelline.

In questa parte di codice 4.7 possiamo vedere come le stelline vengano accese in base al valore del punteggio finale dell'esercizio. Sono state identificati in maniera arbitraria degli intervalli di punteggio sui quali fare il controllo, questi possono essere semplicemente modificati, andando a cambiare i valori degli stessi.

#### 4.4.4 AR

Con l'inserimento della scena AR vengono creati automaticamente due classi: la `DefaultInitializationErrorHandler` e la `DefaultTrackableEventHandler`. La prima funge semplicemente da gestore degli errori, come si evince dal nome; mentre la seconda serve a gestire gli eventi che avvengono nella scena AR. Tre sono i metodi utilizzati: `OnTrackingFound`, `OnTrackingLost` e `OnTrackableStateChanged`.

Come possiamo vedere nel metodo `OnTrackingFound` 4.8 vengono settati a “TRUE” il `Renderer` che permette appunto di renderizzare l'oggetto 3D, il `Collider` che permette agli oggetti di interagire fisicamente tra loro e il `Canvas` cioè la finestra dove verrà effettuato il render. A queste funzioni, che sono di default, è stato aggiunto il metodo `Rewind()` su tutte le animazioni presenti su un determinato marker. Questo fa sì che, se l'utente “perde” il marker, ogni volta che questo verrà di nuovo identificato l'animazione partirà dall'inizio. Inoltre, in Unity 3D, le animazioni sono state salvate con “Wrap Mode” loop, così che l'animazioni sia visibile più volte senza il bisogno di staccare l'inquadratura del marker e riprenderla.

---

```
1
2 protected virtual void OnTrackingFound()
3     {
4
5         var rendererComponents =
6             GetComponentInChildren<Renderer>(true);
7         var colliderComponents =
8             GetComponentInChildren<Collider>(true);
9         var canvasComponents =
10             GetComponentInChildren<Canvas>(true);
11
12         // Enable rendering:
13         foreach (var component in rendererComponents)
14             component.enabled = true;
15
16         Animation[] array =
17             mTrackableBehaviour.GetComponentInChildren<Animation>(
18                 false);
19         foreach (Animation r in array)
20         {
21             r.Rewind();
22         }
23
24         // Enable colliders:
25         foreach (var component in colliderComponents)
26             component.enabled = true;
27
28         // Enable canvas':
29         foreach (var component in canvasComponents)
30             component.enabled = true;
```

26  
27           }

---

Elenco dei codici 4.8: Metodo `OnTrackingFound()` richiamato ogni volta che la camera individua un marker.

Il metodo `OnTrackingLost` 4.9 fa l'opposto del metodo precedente: scorre tutti i `Component` e i loro figli che sono "TRUE" e li setta a "FALSE". Questo viene richiamato ogni volta che la camera perde la visualizzazione di un marker.

---

```
1
2 protected virtual void OnTrackingLost()
3     {
4         var rendererComponents =
5             GetComponentInChildren<Renderer>(true);
6         var colliderComponents =
7             GetComponentInChildren<Collider>(true);
8         var canvasComponents =
9             GetComponentInChildren<Canvas>(true);
10
11         // Disable rendering:
12         foreach (var component in rendererComponents)
13             component.enabled = false;
14
15         // Disable colliders:
16         foreach (var component in colliderComponents)
17             component.enabled = false;
18
19         // Disable canvas':
20         foreach (var component in canvasComponents)
21             component.enabled = false;
22     }
```

---

Elenco dei codici 4.9: Metodo `OnTrackingLost()` richiamato ogni volta che la camera perde un marker.

Molto più interessante è il metodo `OnTrackableStateChanged` 4.10. Questo metodo si occupa di gestire i cambiamenti che avvengono durante l'inquadratura di un marker. Esso ha due stati "previousStatus" (prima dell'individuazione di un marker) e "newStatus" (dopo che è stato individuato il marker). Si possono avere tre casi: il primo in cui il nuovo stato è "marker found" che viene definito dagli attributi "DETECTED", "TRACKED" o "EXTENDED\_TRACKED". In questo caso viene richiamato il metodo `OnTrackingFound` e a seconda del nome del marker, definito in Unity 3D, viene caricata la scena di tipo "game" ad esso associata. Negli altri due casi in cui: lo stato precedente sia "marker found"

mentre quello nuovo sia “marker lost” o siano entrambi “marker lost” viene richiamato il metodo OnTrackingLost.

---

```
1
2 public void OnTrackableStateChanged(
3         TrackableBehaviour.Status previousStatus,
4         TrackableBehaviour.Status newStatus)
5     {
6         _dataController = FindObjectOfType<DataController> ();
7         if (newStatus == TrackableBehaviour.Status.DETECTED ||
8             newStatus == TrackableBehaviour.Status.TRACKED ||
9             newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
10    {
11
12        OnTrackingFound();
13        if (_dataController.GetSizeOfGameByRound
14            (_dataController.round_active) >
15            _dataController.getGameCount ()) {
16
17
18            if (mTrackableBehaviour.TrackableName ==
19                "marker-1") {
20
21                _dataController.increaseGameCount ();
22                SceneManager.LoadScene ("Game");
23
24            } else if (mTrackableBehaviour.TrackableName ==
25                "marker-1_1") {
26
27                SceneManager.LoadScene ("Game1");
28
29            } else if (mTrackableBehaviour.TrackableName ==
30                "marker-1_2") {
31
32                _dataController.increaseGameCount ();
33                SceneManager.LoadScene ("Game1");
34
35            }else if(mTrackableBehaviour.TrackableName ==
36                "marker-1_3"){
37
38            }else if (mTrackableBehaviour.TrackableName ==
39                "marker-2") {
40
41                _dataController.increaseGameCount ();
42                SceneManager.LoadScene ("Game2");
43
44            }
45        }
46    }
```

```
40         } else if (mTrackableBehaviour.TrackableName ==  
41             "marker-2_1") {  
42             _dataController.increaseGameCount ();  
43             SceneManager.LoadScene ("Game3");  
44         }  
45         } else if (mTrackableBehaviour.TrackableName ==  
46             "marker-3") {  
47             _dataController.increaseGameCount ();  
48             SceneManager.LoadScene ("Game4");  
49         }  
50         } else if (mTrackableBehaviour.TrackableName ==  
51             "marker-3_1") {  
52             _dataController.increaseGameCount ();  
53             SceneManager.LoadScene ("Game5");  
54         }  
55         } else if (mTrackableBehaviour.TrackableName ==  
56             "marker-4") {  
57             _dataController.increaseGameCount ();  
58             SceneManager.LoadScene ("Game6");  
59         }  
60         } else if (mTrackableBehaviour.TrackableName ==  
61             "marker-4_1") {  
62             _dataController.increaseGameCount ();  
63             SceneManager.LoadScene ("Game7");  
64         }  
65         } else if (mTrackableBehaviour.TrackableName ==  
66             "marker-5") {  
67             _dataController.increaseGameCount ();  
68             SceneManager.LoadScene ("Game8");  
69         }  
70         } else if (mTrackableBehaviour.TrackableName ==  
71             "marker-5_1") {  
72             _dataController.increaseGameCount ();  
73             SceneManager.LoadScene ("Game9");  
74         }  
75         } else if (mTrackableBehaviour.TrackableName ==  
76             "marker-6") {  
77             _dataController.increaseGameCount ();  
78             SceneManager.LoadScene ("Game10");  
79         }
```



```
80         } else if (mTrackableBehaviour.TrackableName ==  
81             "marker-6_1") {  
82             _dataController.increaseGameCount ();  
83             SceneManager.LoadScene ("Game11");  
84         }  
85         } else if (mTrackableBehaviour.TrackableName ==  
86             "marker-7") {  
87             _dataController.increaseGameCount ();  
88             SceneManager.LoadScene ("Game12");  
89         }  
90         } else if (mTrackableBehaviour.TrackableName ==  
91             "marker-8") {  
92             _dataController.increaseGameCount ();  
93             SceneManager.LoadScene ("Game13");  
94         }  
95         } else if (mTrackableBehaviour.TrackableName ==  
96             "marker-9") {  
97             _dataController.increaseGameCount ();  
98             SceneManager.LoadScene ("Game14");  
99         }  
100     }  
101     } else if (mTrackableBehaviour.TrackableName ==  
102         "marker-10") {  
103         _dataController.increaseGameCount ();  
104         SceneManager.LoadScene ("Game15");  
105     }  
106     } else if (mTrackableBehaviour.TrackableName ==  
107         "marker-11") {  
108         _dataController.increaseGameCount ();  
109         SceneManager.LoadScene ("Game16");  
110     }  
111     } else if (mTrackableBehaviour.TrackableName ==  
112         "marker-12") {  
113         _dataController.increaseGameCount ();  
114         SceneManager.LoadScene ("Game17");  
115     }  
116     } else if (mTrackableBehaviour.TrackableName ==  
117         "marker-13") {  
118         _dataController.increaseGameCount ();  
119         SceneManager.LoadScene ("Game18");  
120     }
```

```
121
122         } else if (mTrackableBehaviour.TrackableName ==
123             "marker-14") {
124             _dataController.increaseGameCount ();
125             SceneManager.LoadScene ("Game19");
126
127             }
128
129         } else {
130
131             SceneManager.LoadScene
132                 ("MenuScreen");
133
134         }
135         else if (previousStatus ==
136             TrackableBehaviour.Status.TRACKED &&
137             newStatus ==
138                 TrackableBehaviour.Status.NOT_FOUND)
139         {
140
141             OnTrackingLost();
142
143         }
144         else
145         {
146
147             OnTrackingLost();
148
149         }
150     }
```

---

Elenco dei codici 4.10: Metodo OnTrackableStateChanged() richiamato ogni volta che c'è un cambiamento nello stato dell'oggetto tracciato.

Un'altra classe fondamentale che è stata creata per questa applicazione specifica è NOARCamera 4.11. Con Unity 3D e Vuforia ogni qual volta viene aggiunta una camera AR questa rimane attiva per tutto il flusso dell'applicazione. Nel caso specifico di ARhino questo avrebbe comportato la sovrapposizione della parte di realtà aumentata all'interfaccia 2D delle domande rendendo il tutto poco leggibile e avrebbe compromesso il corretto l'utilizzo dell'app. Questa classe permette di “spegnere” la camera AR durante il flusso delle domande disabilitando tutti gli attributi presenti sulla camera: in particolare VuforiaBehavior, VideoBackGroundBehaviour e DefaultInitializationErrorHandler.

```
1
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using Vuforia;
6
7 public class NOARCamera : MonoBehaviour {
8
9 void Start () {
10
11 Camera mainCamera = Camera.main;
12 if( mainCamera )
13     {
14         if( mainCamera.GetComponent<VuforiaBehaviour>( ) != null )
15         {
16             mainCamera.GetComponent<VuforiaBehaviour>().enabled =
17                 false;
18         }
19         if( mainCamera.GetComponent<VideoBackgroundBehaviour>( )
20             != null )
21         {
22             mainCamera.GetComponent<VideoBackgroundBehaviour>().enabled
23                 = false;
24         }
25         if(
26             mainCamera.GetComponent<DefaultInitializationErrorHandler>(
27                 ) != null )
28         {
29             mainCamera.GetComponent<DefaultInitializationErrorHandler>().enabled
30                 = false;
31         }
32     }
33 }
```

---

Elenco dei codici 4.11: Classe NOARCamera realizzata per disabilitare la camera AR.

## Capitolo 5

# Descrizione Esercizi

### 5.1 Gli Esercizi

Come accennato nel capitolo precedente ARhino si compone di un numero limitato di “livelli”, che hanno come fine ultimo quello di facilitare l’apprendimento dei passaggi più significativi relativi agli esercizi, già utilizzati nelle slides del corso di studi. Le domande per gli esercizi sono poste in inglese poiché questa applicazione è rivolta ad un corso di studi seguito da studenti internazionali, vedi (Appendice A). Di seguito verrà descritto nel dettaglio il primo esercizio come esempio per evidenziare la struttura generale degli esercizi realizzati nell’applicazione. Subito dopo verrà fatta una panoramica generale su tutti gli altri, soffermandosi sui motivi della scelta di evidenziare alcuni punti rispetto ad altri e le domande per raggiungere questo scopo.

#### 5.1.1 Esercizio 1

Come possiamo vedere nel diagramma (Figura 5.1), nella prima fase di ogni esercizio lo studente segue le slides relative al corso iniziando a svolgere l’esercizio (box nero). Nelle slide compare il primo marker, con una didascalia che invita ad usare l’app ARhino. Lo studente apre l’app e la prima schermata che si trova davanti è quella dell’introduzione in cui viene descritta l’app e vengono spiegate regole di utilizzo (Figura 4.1). Dopo aver letto le istruzioni e cliccato sul bottone “CLOSE” allo studente apparirà la schermata del menu dal quale è possibile, accedere alla schermata degli esercizi, in cui ad ogni bottone corrisponde un livello (Figura 4.2).

Cliccando sul primo bottone verrà aperta la camera AR (primo box giallo), finché non verrà inquadrato o riconosciuto il primo marker la camera rimarrà aperta. Appena il marker viene riconosciuto, si carica il primo “Game” (frece nere) che contiene le prime due domande, viene fatto partire il timer e il calcolo del punteggio. All’inizio viene caricata la prima domanda (Figura 5.1) (Which command do you use to make the shape in the slide?) con le relative opzioni. Se lo studente risponde in maniera sbagliata questo viene segnalato con un’icona in alto a sinistra di colore rosso e viene inserito un messaggio che esorta a riprovare (Figura 5.2 [b]); viene inoltre caricata di nuovo la stessa domanda, però con le opzioni in un ordine differente dal precedente.

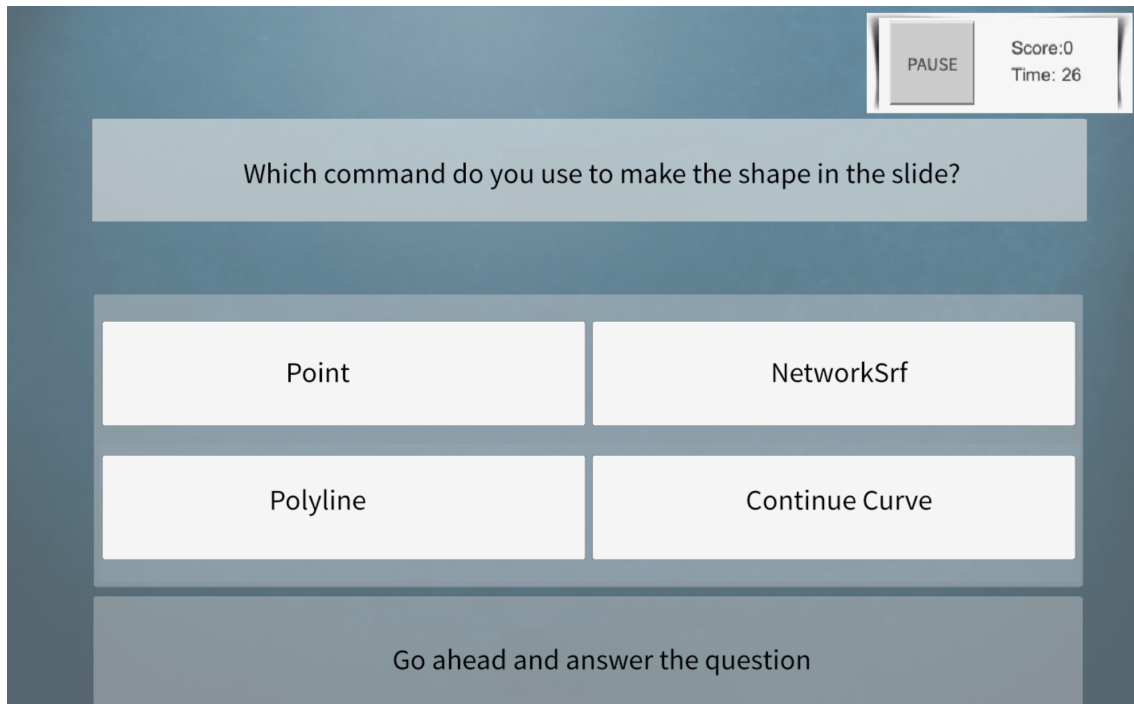
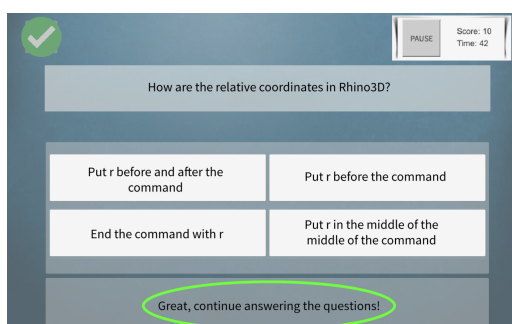
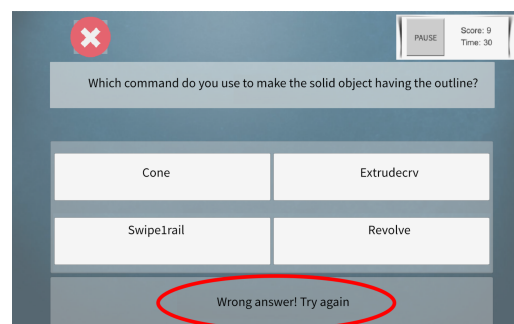


Figura 5.1: Schermata domanda (in particolare la prima domande del primo esercizio).

Se invece risponde in maniera esatta (Figura 5.2 [a]) viene caricata la seconda domanda (How are the relative coordinates in Rhinoceros?). In questo caso appare un' icona di colore verde e un messaggio che comunica di procedere con le successive domande. Anche in questo caso finché lo studente non avrà risposto correttamente verrà ricaricata la stessa domanda.



(a) Seconda domanda del primo esercizio



(b) Quarta domanda del primo esercizio

Figura 5.2: Schermata risposta corretta [a], risposta sbagliata [b]

Appena l'utente risponde correttamente a tutte e due le domande, viene riaperta la camera AR, lo studente inquadra il secondo marker e partirà la prima animazione (primo box rosso) (Figura 5.3) . In questa animazione viene mostrato come creare la freccia in 2D utilizzando lo strumento polyline. Se l'utente “perde”, per qualsiasi motivo, il marker una volta che questo verrà inquadrato di nuovo l'animazione partirà d'accapo.

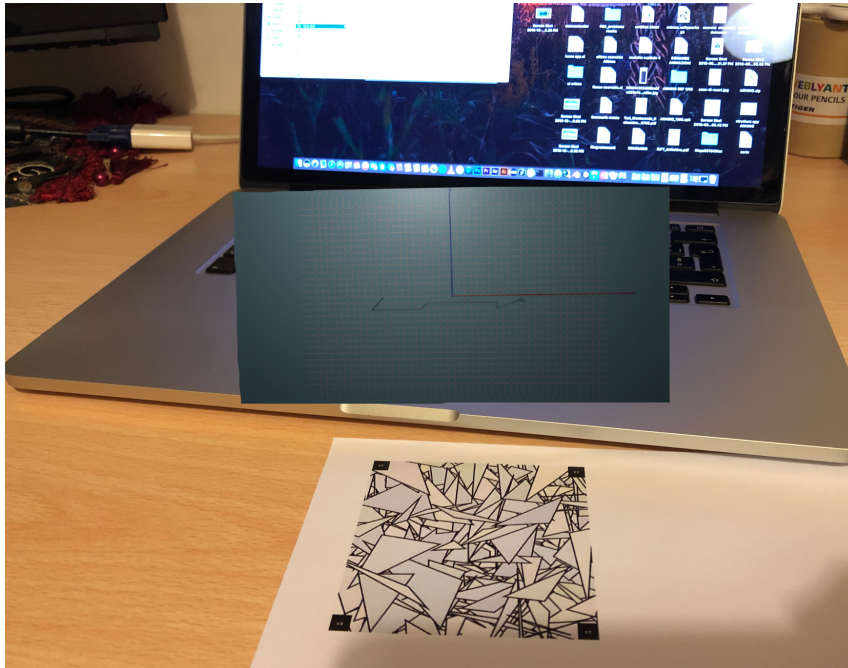


Figura 5.3: Frame della prima animazione del primo esercizio (la prima metà della freccia è stata creata).

Visualizzata l'animazione, l'utente inquadrerà il marker successivo per caricare il secondo "Game" (freccie viola). Appena caricato verrà disabilitata la possibilità di rispondere alle domande e verrà comunicato (attraverso il pannello di comunicazione), che per procedere bisognerà mettere in pausa l'applicazione e provare a riprodurre quanto appena visto, in Rhinoceros sul proprio computer (Figura 5.4).

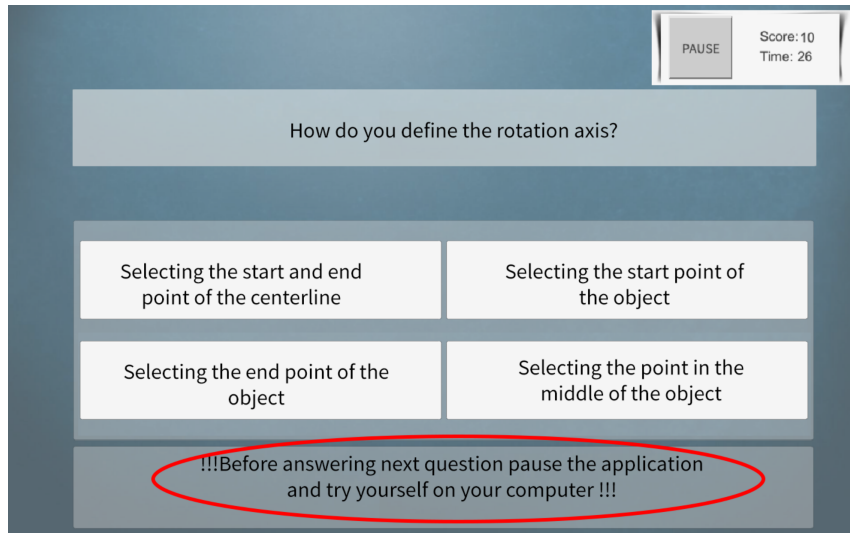


Figura 5.4: Schermata domanda successiva all'animazione: viene richiesto di mettere pausa.

Dopo che l'utente ha schiacciato il bottone "PAUSE", si sposterà sul proprio computer per realizzare il passo appena visto. Una volta completata la prova si ritornerà sull'applicazione e schiacciando il bottone "RESUME GAME" (Figura 4.3) si risponderà alla domanda successiva, in questo caso la terza (How do you define the rotation axis?) con le relative opzioni. Una volta che lo studente ha risposto correttamente, viene caricata la domanda successiva (Which command do you use to make the solid object having the outline?).

Completato il secondo "Game" viene riaperta la camera AR, per inquadrare il marker successivo (frecche verdi nel diagramma). Inquadrato questo marker verrà caricata la seconda animazione (Figura 5.5).

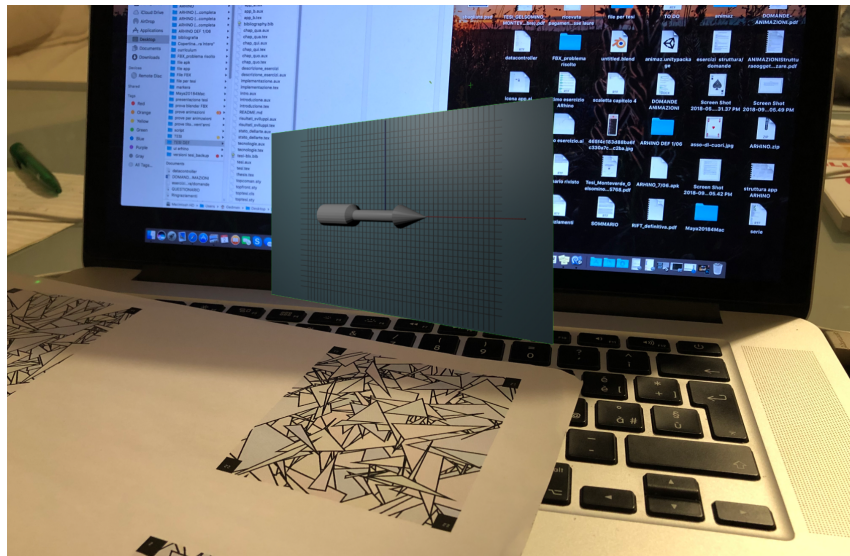


Figura 5.5: Frame della seconda animazione del primo esercizio (si forma la freccia in 3D).

Una volta terminata la visualizzazione lo studente inquadra il marker successivo e come in precedenza viene richiesto di mettere l'app in pausa e provare a realizzare, quanto visto, sul computer. Lo studente dovrà poi rispondere all'ultima domanda che rappresenta il terzo "Game" (It's everything clear, do you understood all the steps of the exercise?). Questa domanda finale (Figura 5.6) è comune a tutti gli esercizi. Se l'utente risponde "YES", verrà aperta la schermata di fine esercizio dove verranno visualizzati punteggio, tempo e l'immagine di un cane (che varia a seconda del punteggio: punteggio alto cane felice, punteggio medio cane normale e punteggio basso cane triste). Mentre se la risposta è "NO" verrà visualizzato un pannello di comunicazione che rimanda al professore per maggiori spiegazioni. Da qui l'utente schiacciando sul bottone "OK" aprirà la schermata di fine esercizio. Il punteggio viene calcolato sommando tutti i valori per le risposte esatte e sottraendo un valore per ogni risposta sbagliata. A questo risultato verrà detratta una percentuale del 20% se il tempo impiegato è al di sopra della seconda soglia di tempo per quell'esercizio, del 10% se il tempo si trova tra le due soglie definite per quell'esercizio.



Mentre se il tempo è minore il punteggio non subirà variazioni.

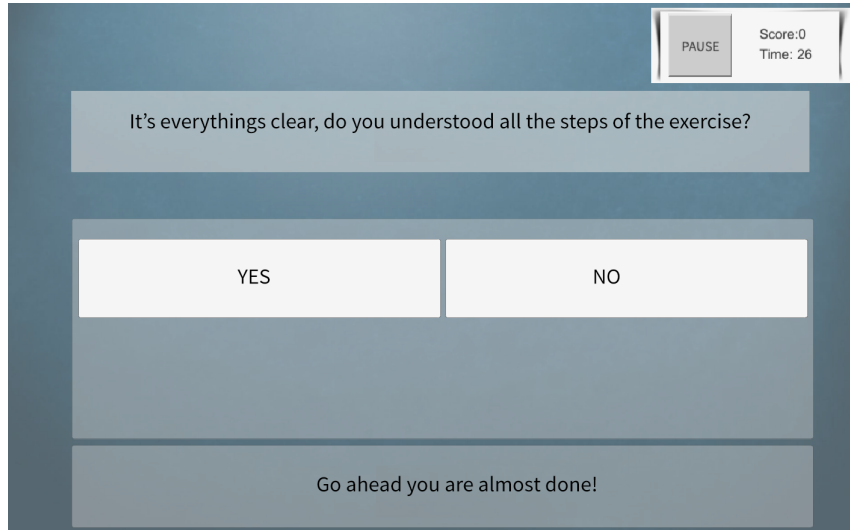


Figura 5.6: Schermata ultima domanda.

Dalla schermata di fine esercizio, infine, cliccando sul bottone “RETURN TO MENU” si tornerà al menu, qui verranno salvati i dati di tempo e punteggio nel bottone relativo all’esercizio appena terminato e verrà attivato il bottone relativo all’esercizio successivo. Il punteggio sarà visualizzato inoltre con delle stelline che vengono attivate in base al valore del punteggio su una scala di cinque intervalli.

### 5.1.2 Esercizio 2

74

### 5.1.3 Esercizio 3

L'esercizio 3, come l'esercizio precedente si compone di una sola animazione, ma in questo caso si hanno due domande. Nella prima domanda viene richiesto qual è il comando da utilizzare per creare l'oggetto nella slide. In questo caso la domanda viene posta in maniera negativa così da rendere meno immediata la risposta e indurre lo studente al ragionamento.

Alla fine dell'animazione relativa a questa domanda, nel panel di comunicazione, compare una nota. Questa precisa che per realizzare l'oggetto sono stati utilizzati determinati comandi, ma che si può arrivare allo stesso risultato utilizzando comandi diversi.

Nella seconda domanda viene richiesto il motivo dell'uso dello "snap" all'oggetto. Questo comando aiuta a scegliere un punto preciso su un oggetto come per esempio la fine di una linea o il centro di un cerchio. Anche in questo caso lo studente dovrà mettere in pausa l'applicazione ed eseguire l'oggetto in Rhino3D.

### 5.1.4 Esercizio 4

Nell'esercizio 4 vengono approfonditi due concetti molto importanti: l'estrusione di un oggetto piano e i booleani.

Nella prima domanda viene richiesto di definire il comando che permette di creare lo spessore di un oggetto piano ed in seguito viene visualizzato il passaggio tramite un'animazione. Dopo che lo studente ha realizzato il primo passaggio sul computer, si passa alla seconda parte dell'esercizio dove viene richiesto come si fa per ottenere il solido "bucato" e come fare per non cancellare gli oggetti usati per creare la cavità.

Questo esercizio serve a far capire come un oggetto 3D possa essere realizzato non soltanto come somma di più oggetti, ma come sottrazione di una parte dell'oggetto. Anche qui, alla fine delle domande, viene mostrata l'animazione relativa alla "differenza" e lo studente è invitato a provare a realizzare l'oggetto sul proprio computer.

### 5.1.5 Esercizio 5

L'esercizio 5 o esercizio del "cacciavite" è diviso in due parti, come visto per l'esercizio 1: nella prima parte viene creato l'oggetto 2D e nella seconda quello 3D.

La prima domanda riguarda la creazione di diversi layer sui quali andare a disegnare oggetti differenti, in questo caso un layer di "costruzione", che funge da guida, e un layer "curve" che invece è l'oggetto che si vuole realizzare. Le due domande successive richiedono informazioni sui comandi da utilizzare per realizzare le figure del layer di "costruzione" e quello "curve". Dopo aver risposto alle domande verrà fatta vedere l'animazione relativa alla creazione della guida e del contorno del cacciavite. Conclusa l'animazione, viene inserita nel pannello di comunicazione una nota che evidenzia come usando più punti di controllo si possano ottenere curve via via più precise.

Dopo che lo studente ha realizzato la parte dell'oggetto sul suo computer, si passa alla seconda parte. Questa si compone di due domande: una su come realizzare l'asse di rotazione e l'altra su quale comando utilizzare per creare il solido a partire dall'oggetto realizzato

nella prima parte. In questo caso sono state usate volutamente le stesse domande dell'esercizio 1, con relativa animazione, per far sì che lo studente memorizzi il processo anche in maniera ripetitiva.

### 5.1.6 Esercizio 6

L'esercizio 6 o esercizio "Gasket" è simile all'esercizio 3 ed è composto da due animazioni e quattro domande. La prima domanda richiede di indicare quale comando non utilizzare per creare l'oggetto nella slide. Anche in questo caso viene posta la domanda in modo negativo, sia perchè come in precedenza non c'è un unico comando per creare l'oggetto e sia per invogliare lo studente a ragionare. La seconda domanda richiede allo studente quale sia il comando da utilizzare per unire i vari archi realizzati. Alla fine di queste due domande e alla relativa animazione viene inserita una nota nel pannello comunicazione, nella quale vengono indicati i comandi utilizzati per realizzare gli oggetti nell'animazione e viene sottolineato che ne possono essere utilizzati altri per realizzare lo stesso oggetto. Dopo che l'utente ha svolto i passaggi relativi alla prima animazione sul suo computer, si passa alla seconda parte in cui viene realizzato l'oggetto con un determinato spessore. La prima domanda relativa a questa parte è come si può realizzare un oggetto singolo da tutti gli oggetti realizzati che lo compongono, la seconda invece richiede come sia possibile trasformare l'oggetto in un oggetto con un determinato spessore.

### 5.1.7 Esercizio 7

L'esercizio 7 o esercizio del "bicchiere" è diviso in due parti: nella prima viene richiesto qual è il comando per realizzare il profilo del bicchiere e, attraverso la definizione dell'asse di rotazione, il solido (come negli esercizi 1 e 5), nella seconda parte il solido viene modificato in modo da creare un bicchiere con un bordo non lineare. La domanda della seconda parte chiede come è possibile modificare la forma del solido ottenuto nella prima parte. La relativa animazione invece, rappresenta come, utilizzando i punti di controllo attivati sull'oggetto, è possibile modificare facilmente la forma dell'oggetto.

### 5.1.8 Esercizio 8

L'esercizio 8 o esercizio della "paperella" è quello che contiene più animazioni, in quanto sono diverse le parti da realizzare per poter creare una "paperella". Nella prima fase viene creato il corpo attraverso due sfere, la prima domanda è appunto come realizzare una sfera, mentre la seconda richiede quale sia l'operatore che si usa per ricostruire una superficie. Ovvero ricostruire le superfici selezionate secondo il grado ed il numero di punti di controllo. Alla fine di queste due domande verrà visualizzata l'animazione relativa alla realizzazione del corpo e della testa della "paperella".

Una volta che lo studente ha realizzato i passaggi per creare il corpo e la testa sul suo computer si passa alle domande successive relative alla deformazione delle sfere per rendere la "paperella" più realistica. In primo luogo viene richiesto qual è la modalità da utilizzare per settare i punti di controllo e viene subito visualizzata l'animazione della deformazione delle sfere.

Nella parte successiva viene richiesto come realizzare la curva che serve per dividere il becco dalla testa e l'operatore da utilizzare per dividere le due sezioni. Dopo queste due domande comparirà l'animazione della divisione del becco.

Nel passaggio successivo verrà richiesto di definire l'operatore per unire le due sezioni e verrà visualizzata la relativa animazione. In seguito si passa alla realizzazione del collo, questa fase si divide in due parti la prima in cui la testa viene tagliata nella parte inferiore e la seconda in cui avviene la creazione vera e propria del collo. Nella prima parte viene chiesto cosa usare come piano di taglio per secare la parte inferiore della testa della "paperella" e, nella domanda successiva, qual è il comando da utilizzare per tagliare la parte di sfera. A queste domande segue l'animazione della definizione del piano e del taglio.

Nella seconda parte viene richiesto, a partire dalla superficie creata con il taglio, come creare la superficie del collo, cosa si può utilizzare per tagliare il collo e il comando per creare la superficie per unire il collo alla testa. Alla fine di queste tre domande viene visualizzata l'ultima animazione che riassume la realizzazione del collo della "paperella".

### 5.1.9 Esercizio 9

L'esercizio 9 è composto da sei animazioni e si suddivide in altrettante fasi. La prima parte consta nella realizzazione del box: viene richiesto quale operatore usare per la creazione dell'oggetto nella slide, come si fa per estrarre una superficie da un oggetto solido e fare il rebuild della superficie specificando il numero di punti di controllo. Nella prima animazione viene descritta la realizzazione del box. Come sempre viene richiesto allo studente di realizzare lo stesso oggetto sul computer prima di andare avanti con le domande successive.

Nella seconda parte viene richiesto come modificare la forma dell'oggetto, come avvenuto nell'esercizio del "bicchiere" e della "paperella", e come fare per unire le due superfici. Compare quindi l'animazione che deforma l'oggetto in modo da renderlo concavo.

Nella terza parte il solido fino ad ora realizzato, viene "chiuso" cioè vengono unite le superfici separate. Viene chiesto quale strumento utilizzare per chiudere il solido e come fare per eliminare i bordi "affilati" dell'oggetto. A queste domande segue un'animazione dove vengono chiusi i bordi e resi lisci.

Le ultime tre parti/animazioni riguardano l'inserimento di testo. All'inizio viene richiesto come inserire del testo in scena e quale strumento usare per creare il testo solido; a queste domande segue l'animazione in cui viene mostrato la creazione del testo.

Nelle ultime due parti viene chiesto come creare un'incisione o un rilievo con il testo e ne vengono mostrate le relative animazioni.

### 5.1.10 Esercizio10

L'esercizio 10 è quello della "cornetta del telefono". In questo esercizio vengono mostrato le diverse modalità di estrusione presenti in Rhino3D: estrusione normale, estrusione lungo una curva ed estrusione verso un punto.

Nella prima domanda viene richiesto quale comando viene usato per creare una figura dritta come nella slide. Nella relativa animazione è rappresentata una semplice estrusione

lineare.

Nella seconda domanda, invece viene richiesto quale comando usare per creare una figura non lineare avendo una curva come guida. Nell'animazione relativa viene riprodotta l'estrusione lungo una curva.

Infine viene richiesto come realizzare la parte concava presente nella slide e l'animazione successiva rappresenta l'estrusione verso un punto.

Questo esercizio permette di evidenziare i vari metodi che esistono in Rhino3D per l'estrusione e di collegarli alla realizzazione di un unico oggetto.

#### **5.1.11 Esercizio 11**

L'esercizio 11 o esercizio "canoa" è composto da due animazione. Nella prima parte viene chiesto quale comando usare per creare le curve di costruzione e subito nell'animazione successiva vengono create le curve. Alla fine dell'animazione nel pannello di comunicazione viene evidenziato come le curve debbano avere tutte lo stesso rango per far sì che non ci siano errori, quindi con lo stesso numero di punti di controllo.

Nella seconda parte viene chiesto quale comando usare per realizzare la superficie avendo le curve di costruzione. Nell'animazione relativa a questa parte viene visualizzata l'animazione del loft che permette di realizzare una metà della canoa. Infine dopo che lo studente ha realizzato i due passaggi precedenti sul computer, si passa all'ultima parte nella quale viene richiesto come realizzare la canoa completa avendo la metà creata in precedenza. Quindi nell'ultima animazione viene rappresentato il mirror e la creazione dell'oggetto completo.

#### **5.1.12 Esercizio 12**

L'esercizio 12 o esercizio del "cuore" è il primo degli ultimi tre che hanno una domanda e un'animazione ciascuno. In questo esercizio viene introdotto il comando di rivoluzione. Nella domanda viene richiesto quale comando utilizzare per realizzare un solido avendo una curva di percorso. Nell'animazione relativa viene mostrata la formazione del "cuore" con il comando rivoluzione.

#### **5.1.13 Esercizio 13**

Nell'esercizio 13 viene introdotto il comando di rivoluzione su un binario. Per realizzare questo esercizio e quello precedente vengono utilizzati dei file già realizzati con le linee da usare come guide.

In questo esercizio viene richiesto quale comando utilizzare per realizzare un solido avendo una curva che funge da binario. Nell'animazione relativa viene mostrata la formazione del solido che segue la linea usata come binario.

#### **5.1.14 Esercizio 14**

Nell'ultimo esercizio, il numero 14, viene analizzato invece il comando di rivoluzione su due binari. Questo fa parte di un esercizio più ampio che permette di realizzare uno specchietto

retrovisore. Questo passaggio rappresenta la prima parte in cui viene creata la base dello specchietto. Anche in questo caso di parte da una file già presente che contiene le curve da usare come binari. La domanda è simile a le due precedenti: quale operatore usare per creare l'oggetto nelle slide avendo le due curve come binari.

## Capitolo 6

# Conclusioni e Sviluppi Futuri

### 6.1 Conclusioni

L'obiettivo della tesi è stato quello di sviluppare un'applicazione in realtà aumentata che permettesse di rendere più intuitivi e meno complessi alcuni passaggi della modellazione 3D e quindi facilitarne l'apprendimento. L'applicazione è stata progettata per dispositivi mobile/tablet Android in quanto sono la piattaforma perfetta per questo tipo di applicazione perché sono facili da trasportare, sono già presenti sul mercato e sono economicamente vantaggiosi. Inoltre, questi dispositivi integrano tutti i diversi tipi di sensori di cui ha bisogno un sistema di realtà aumentata. Per quanto riguarda il sistema operativo si è scelto di usare Android poiché è attualmente il più diffuso.

Durante lo sviluppo e la progettazione dell'applicazione, sono state diverse le problematiche e le difficoltà da dover superare, soprattutto relative ai software utilizzati per la realizzazione. Tra le sfide principali vi è stata sicuramente quella riguardante l'unione di un'interfaccia 2D ed un ambiente 3D della realtà aumentata, o ancora la difficoltà di realizzare con un software blender i passaggi propri di un altro software come Rhino3D per realizzare gli stessi oggetti. Infine ma non meno importante il passaggio delle animazioni da Blender a Unity. È chiaro che una rappresentazione fedele delle animazioni è impossibile da ottenere quando si cerca di realizzare con un altro software in questo caso Blender le animazioni di come vengono realizzati gli oggetti in Rhino3D. Per tale ragione è stato necessario uno sforzo maggiore e tempi più lunghi per ottimizzare le animazioni e far sì che queste vengano ben visualizzate nel software Unity3D.

Questa applicazione è stata realizzata con lo scopo specifico di aiutare gli studenti del corso di "Design di Scenario" nell'apprendimento di concetti che possono risultare ostici. Si tratta di un corso di studi che viene effettuato durante il primo semestre dell'anno accademico, che va da settembre a Dicembre. Essendo quindi, il corso quasi giunto al termine, si prevede che l'applicazione venga testata l'anno prossimo in modo da poter avere risultati più veritieri considerando che sarà possibile far utilizzare l'applicazione fin dai primi giorni del corso.

L'applicazione verrà fornita all'inizio del corso come supporto alle slides dell'insegnamento.



La classe verrà divisa in due gruppi eterogenei (maschi e femmine): il gruppo sperimentale e il gruppo di controllo. Soltanto la primo verrà data la possibilità di utilizzare l'applicazione in aggiunta alle slides, mentre il secondo, ovvero gruppo di controllo seguirà il corso come si è fatto fino ad oggi con le slide come unico aiuto. Alla fine del corso agli studenti del gruppo sperimentale verrà sottoposto un questionario per valutare le loro opinioni riguardo l'utilizzo di questa applicazione come integrazione al normale materiale di corso, inoltre verrà fatta una comparazione tra i voti dei ragazzi presenti nei due gruppi, per analizzare se oltre ad essere qualcosa di bello e usabile porta anche dei miglioramenti nell'apprendimento e quindi nel rendimento in termini di voti.

## 6.2 Sviluppi Futuri

Il questionario (Appendice B) che verrà sottoposto agli studenti alla fine del corso, dopo aver quindi utilizzato l'applicazione per un semestre, servirà anche come base informativa per valutare i miglioramenti e le criticità riscontrate. È possibile in via preventiva ipotizzare dei possibili sviluppi futuri che hanno lo scopo di aumentare l'efficacia di questo strumento. Innanzitutto si può pensare di migliorare la qualità delle animazioni valutando la possibilità di utilizzare delle tecnologie innovative e poco conosciute che lo permettano. A livello strutturale si può pensare di aumentare il numero di esercizi in base alle necessità e anche sulla base dei feedback ricevuti dagli studenti. Questa modifica è facilmente realizzabile grazie alla struttura scalabile del progetto.

Per quanto riguarda il contenuto è possibile ampliare le funzionalità dell'applicazione creando un ambiente "social" per la condivisione delle informazioni. Ad esempio è possibile prevedere un gruppo chiuso su Facebook dove i ragazzi possono aiutarsi dando dei consigli gli uni agli altri e postando qualsiasi problema relativo all'applicazione. Questa sorta di collaborazione potrebbe essere estesa anche a dei laboratori dove effettuare delle prove intermedie andando a risolvere degli esercizi aggiuntivi in gruppi di due. Un altro campo in cui si possono aggiungere elementi è quello della gamification, infatti oltre alla ricompensa simbolica attuale delle stelline e della possibilità di sbloccare il nuovo esercizio, si potrebbe pensare a delle ricompense reali come per esempio dei punti aggiuntivi al voto finale se si realizzano tutti gli esercizi con determinati punteggi o nel portare a termine dei compiti settimanali o bi-settimanali anche in gruppi di due.

# Appendice A

## Domande Esercizi

### ESERCIZIO 1

[1] Which command do you use to make the shape in the slide?

1. Polyline
2. Point
3. NetworkSfr
4. Continue curve

[2] How are the relative coordinates in Rhino3D?

1. Put r before the command
2. Put r before and after the command
3. End the command with r
4. Put r in the middle of the middle of the command

[3] How do you define the rotation axis?

1. Selecting the start and end point of the centerline
2. Selecting the point in the middle of the object
3. Selecting the end point of the object
4. Selecting the start point of the object

[4] Which command do you use to make the solid object having the outline?

1. Revolve
2. Cone
3. Sweep1rail
4. Extrudecrv

### ESERCIZIO 2

[1] How do you set a new construction plane?

1. Set C Plane
2. Click in the perspective view
3. EnableClipping Plane
4. Ground Plane

### ESERCIZIO 3

[1] Which command is not useful to make the shape in the slide?

1. Circle 3 points
2. Circle, center, radius
3. Arc start, end, point
4. box

NOTE: In the last animation have been used 3 commands: arc( center, start, angle), line, arc( start, end, radius), but you can use other commands to achieve the same results.

[2] Why is it important to use the object snap in this case/exercise?

1. To correctly join parts of profile
2. To get the profile faster
3. To get the same reference point
4. To use only the arc command

#### ESERCIZIO 4

[1] Which command do you use to get the hole from the second object?

1. Difference
2. Variableblendsrf
3. Intersect
4. Offsetsrf

[2] How to retain both the solids used to make the difference?

1. Delete input=no
2. Delete input=yes
3. Booleanunion
4. Intersect

#### ESERCIZIO 5

[1] How to create two different layers?

1. Creating a new viewport
2. Activating a level
3. Booleanunion
4. Changing object level

[2] Which command use to make the shape in the construct layer?

1. Polyline
2. Estend curv
3. Network curv
4. Freeform controlpoints

[3] Which command use to make the shape in the curve layer?

1. Freeform controlpoints
2. Line
3. Polyline
4. Extendrvonsrf

[4] How do you define the rotation axis?

1. Selecting the start and end point of the centerline
2. Selecting the point in the middle of the object
3. Selecting the end point of the object
4. Selecting the start point of the object

[5] Which command use to make the solid?

1. Revolve
2. Swipe2rails
3. Extrude
4. Union

### ESERCIZIO 6

[1] Which command it's useless to make the shape shown?

1. Arc (center, start, angle)
2. Polyline
3. Circle
4. Arc (start, end,diameter)

[2] Which command can be used to make the lines that connect the main arcs?

1. Fillet
2. Chamfer
3. Extendcrv
4. Offsetcrvonsrf

[3] Once you created all the arcs how can make the pieces a single object?

1. Join
2. Connectsrf
3. Revolve
4. Extendsrf

[4] How to make the shape a solid object with a specific thickness?

1. Extrude specifying distance
2. Offset specifying distance
3. Rib specifying distance
4. Extrude specifying ext. distance

### ESERCIZIO 7

[1] Which command to use to make the object outline?

1. Freeform controlpoints
2. Extendcrv
3. Curvefromobject (silhouette)
4. Polyline

[2] How do you define the rotation axis?

1. Selecting the start and end point of the centerline

2. Selecting the point in the middle of the object
3. Selecting the end point of the object
4. Selecting the start point of the object

[3] Which command use to make the outline a 3D object?

1. Revolve
2. Rib
3. Extrudecrv
4. Swipe2rails

[4] How to change the shape of the obtained solid?

1. Activating Control points and move them
2. Using fillet edge
3. Using rib
4. Using extrudecrvalongcrv

### ESERCIZIO 8

[1] How to create a sphere?

1. Solid-sphere
2. Extractsrf
3. Ellisspoid
4. Mesh-sphere

[2] Which operator do you use to reconstruct the surface?

1. Rebuild
2. Set point

3. Offset surface
4. Extract surface

[3] How to set the control points?

1. Set points
2. Show points
3. Move
4. Insert knot

[4] What do you use to create the curve?

1. Freeform control points
2. Curve from object (silhouette)
3. Polyline
4. Extend curve

[5] Which operator do you use to divide?

1. Spilt
2. Blend surface
3. Trim
4. Surface edit tool (untrim)

[6] Which operator do you use the merge?

1. Join
2. Paste
3. Trim
4. Union



[7] What you can use as a clipping plane?

1. Cutting plane
2. Rectangle
3. Plane corner to corner
4. Edge curve

[8] Which command do you use to cut the sphere bottom piece with the created plane?

1. Trim
2. Split
3. Join
4. Cut

9. How to create the neck surface?

1. Surface extrude
2. Blend surface
3. Extend surface
4. NetworkSrf

[10] What do you use to cut the part of the cylinder to make the neck?

1. Trim
2. Cut
3. Split
4. Join

11. Which is the command used to join body and head?

1. Blend surface
2. Extend
3. Fillet
4. Sweep2rail

### ESERCIZIO 9

[1] Which operator do you use to create the object in the slide?

1. Box
2. Poligono mesh primitive (box)
3. Extrude surface
4. Curve network

[2] How can you extract a surface from a solid object?

1. Extract surface
2. Extended
3. Blend surface
4. Curve network

[3] How can a surface be rebuilt with a defined number of control points?

1. Rebuild
2. Insertion
3. Extrude surface
4. Curve network

[4] How do you modify the shape of the object to make it roundish?

1. Activating the control points

2. Using the circle operator
3. Using fillet corners
4. Using offset mesh

[5] Which is the command to unite two or more surfaces?

1. Join
2. Rebuild
3. Using fillet corners
4. Extrude surface

[6] Which tool do you use to close the solid?

1. Cap planar holes
2. Connect curves
3. Extrude planar curve
4. Extract surface

[7] Which tool do you use to eliminate the sharp edges?

1. Fillet
2. Extend curve
3. Patch
4. Intersection

[8] Which tool do you use to create the text?

1. Solid text
2. Boolean split
3. Extrude surface

4. Extend curve

[10] Which is the tool to create an engraving, in the solid, with the text created?

1. Difference
2. Smash
3. Fillet surface
4. Offset

[11] Which is the tool to create a raised text in the solid?

1. Estrude solid
2. Extend curve
3. Offset
4. Torus

#### ESERCIZIO 10

[1] Which command to use to make a straight shape?

1. Extrude straight
2. Rib
3. Swipe2rails
4. Extrudecrvopoint

[2] Which command to use in this case to make the shape?

1. Extrudecrvalongcrv
2. Offset
3. Swipe1rail
4. Extrude straight

[3] Which command to use in order to make the concave side as in the slide?

1. Extrudecrvopoint
2. Extrude straight
3. Rib
4. Swipe1rail

#### ESERCIZIO 11

[1] Which command to use if you have to create the curve?

1. Freeform control points
2. Line segments
3. Polyline
4. Extendcrv

NOTE: The curves must have all the same rank

[2] Once you have the construction curves which command to use to create the surface?

1. Loft
2. Offset
3. Extrude straight
4. Patch

[3] Once you create the first half of the canoe which command to use to make the other half equal?

1. Mirros
2. Revolve
3. Union
4. Extrude straight

ESERCIZIO 12

[1] Once you have the curve path, which is the right command to create the solid?

1. Rail revolve
2. Revolve2rails
3. Extrude straight
4. Loft

ESERCIZIO 13

[1] Which command to use to create the solid object in the slide?

1. Sweep1rail
2. Loft
3. Sweep1rail
4. Revolve

ESERCIZIO 14

[1] Which command to use to create the solid object in the slide?

1. Sweep2rail
2. Loft
3. Sweep1rail
4. Revolve

At the end of every exercise:

It's everything clear, do you understand all the steps of the exercise?

1. Yes
2. No

# Appendice B

## Questionario

### ARhino Questionnaire

Please rate the following features based on your evaluation

1. **Gender**

*Mark only one oval.*

☐

Male

☐

Female

2. **Nationality**

---

3. **Would you consider yourself comfortable with using android?**

*Mark only one oval.*

☐

Yes

☐

No

☐

Maybe

4. **What is your mayor purpose of using android for studying?**

---

**5. Do you use your phone/tablet as a tool for studying?**

*Mark only one oval.*

- ☐ Yes  
☐ No  
☐ Maybe

**6. Would you like to use your phone/tablet to learn at your university?**

*Mark only one oval.*

- ☐ Yes  
☐ No  
☐ Maybe

**7. The system provided me with feedback on what I was working. I was not confused or lost while performing the task.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

**8. The messages that appeared in the application were self-explanatory and I understood what they were trying to explain.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

**9. I was in control of the application all the time.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree



10. **I could easily undo/redo any action if I felt to do it.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

11. **The navigation through the application was easy.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

12. **The application had many errors and/or crashed.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

13. **The option to select the desired function was always clear and available all the time and the icon images helped me to know the appropriate functionality of the available options.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

14. **It was easy to find the desired options at any time.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

15. **The application was visually well-designed.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

16. **If error messages appeared, they were clear in their description and probable steps to recover from it were provided.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

17. **I think that interacting with this application requires a lot of mental effort.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

18. **I thought the amount of information displayed on screen was appropriate.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

19. **I thought that the information displayed on screen was difficult to read.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

20. **I felt that the information display was responding fast enough.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

21. **I thought that the information displayed on screen was confusing.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

22. **I thought the words and symbols on screen were easy to read.**

*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

23. I felt that the display was flickering too much.

Mark only one oval.

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totally agree

24. What features of the augmented-reality application are vague or confusing to you; add comments if any?

---

25. Does the menu operation seem easy to operation or difficult? What makes it that way?

---

26. What other features do you think should be included in the augmented- reality Application?

---

27. What do you like best about the augmented-reality application?

---

28. What part or feature do you like the least?

---

29. Scale the rating as per usefulness of Augmented Reality Application

Mark only one oval.

	1	2	3	4	5	
Lowest	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Highest

30. Would you like to use the augmented-reality application for your studies?

Mark only one oval.

- ☐ Yes  
☐ No  
☐ Maybe

31. Any suggestions to improve the augmented-reality application?

---

# Bibliografia

- [1] Ronald T Azuma. «A survey of augmented reality». In: *Presence: Teleoperators & Virtual Environments* 6.4 (1997), pp. 355–385.
- [2] ARToolKit. *Framework per AR*. 2018. URL: <https://www.hitl.washington.edu/artoolkit/>.
- [3] Wayne Piekarski e Bruce H. Thomas. «ARQuake: the outdoor augmented reality gaming system». In: *Commun. ACM* 45 (2002), pp. 36–38.
- [4] La realtà aumentata. *La sua storia*. 2012. URL: <http://larealtaaumentata.blogspot.com/p/la-sua-storia.html>.
- [5] Marshall McLuhan. *Understanding media: The extensions of man*. McGraw-Hill, 1994.
- [6] Gee James Paul. «Good video games and good learning». In: (2005), pp. 33–37.
- [7] Janet Horowitz Murray. *Hamlet on the Holodeck: the future of narrative in cyberspace*. New York USA: The Free Press, 1997.
- [8] John Dewey. *Esperienza e educazione*. IT: La Nuova Italia, 1967.
- [9] Eric Sheldon Josh Klopfer. «Augmenting your own reality: Student authoring of science-based augmented reality games.» In: (2010), pp. 85–95.
- [10] Chris Dede. «Immersive interfaces for engagement and learning.» In: (2009), pp. 66–69.
- [11] Brown J. S. Collins A. Duguid. «Situated cognition and the culture of learning.» In: (1989), pp. 32–42.
- [12] Marcy P. Driscoll. «Psychology of leaning.» In: (2000).
- [13] Schwartz D.L. Sears D. Bransford J.D. «Efficiency and innovation in transfer.» In: (2005), pp. 1–51.
- [14] David N Perkins, Gavriel Salomon et al. «Transfer of learning». In: *International encyclopedia of education* 2 (1992), pp. 6452–6457.
- [15] Jerome Seymour Bruner et al. *Toward a theory of instruction*. Vol. 59. Harvard University Press, 1966.
- [16] Lev Semenovich Vygotsky. *Mind in society: The development of higher psychological processes*. Harvard university press, 1980.

- [17] Matt Dunleavy e Chris Dede. «Augmented reality teaching and learning». In: *Handbook of research on educational communications and technology*. Springer, 2014, pp. 735–745.
- [18] Hsin-Kai Wu et al. «Current status, opportunities and challenges of augmented reality in education». In: *Computers & education* 62 (2013), pp. 41–49.
- [19] Chih-Ming Chen e Yen-Nung Tsai. «Interactive augmented reality system for enhancing library instruction in elementary schools». In: *Computers & Education* 59.2 (2012), pp. 638–652.
- [20] Mau-Tsuen Yang e Wan-Che Liao. «Computer-assisted culture learning in an online augmented reality environment based on free-hand gesture interaction». In: *IEEE Transactions on Learning Technologies* 7.2 (2014), pp. 107–117.
- [21] Masanori Sugimoto. «A mobile mixed-reality environment for children’s storytelling using a handheld projector and a robot». In: *IEEE Transactions on Learning Technologies* 4.3 (2011), pp. 249–260.
- [22] Hannes Kaufmann, Dieter Schmalstieg e Michael Wagner. «Construct3D: a virtual reality application for mathematics and geometry education». In: *Education and information technologies* 5.4 (2000), pp. 263–276.
- [23] Trien V Do e Jong-Weon Lee. «3darmodeler: a 3d modeling system in augmented reality environment». In: *International Journal of Electrical, Computer, and Systems Engineering* 4.2 (2010), pp. 145–154.
- [24] Manfred Lau et al. «Situating modeling: a shape-stamping interface with tangible primitives». In: *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. ACM. 2012, pp. 275–282.
- [25] Jeff KT Tang et al. «Learning to create 3D models via an augmented reality smartphone interface». In: *Teaching, Assessment, and Learning for Engineering (TALE), 2015 IEEE International Conference on*. IEEE. 2015, pp. 236–241.
- [26] Blender.it. *Community italiana di Blender*. 2018. URL: <https://www.blender.it/>.
- [27] Unity3d.com. *Unity 3D sito ufficiale sezione Learn*. 2018. URL: <https://unity3d.com/learn/>.
- [28] Vuforia.com. *Vuforia sito ufficiale sezione Developer*. 2018. URL: <https://developer.vuforia.com>.
- [29] Gamification.it. *Cos’è la Gamification*. 2018. URL: <http://www.gamification.it/gamification/introduzione-alla-gamification>.